



US006609153B1

(12) **United States Patent**
Salkewicz

(10) **Patent No.:** **US 6,609,153 B1**
(45) Date of Patent: **Aug. 19, 2003**

(54) **DOMAIN ISOLATION THROUGH VIRTUAL NETWORK MACHINES**

(75) Inventor: **William Salkewicz, Los Gatos, CA (US)**

(73) Assignee: **Redback Networks Inc., San Jose, CA (US)**

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/220,413**

(22) Filed: **Dec. 24, 1998**

(51) Int. Cl.⁷ **G06F 15/173**

(52) U.S. Cl. **709/223; 370/352; 370/397**

(58) Field of Search **709/221, 224, 709/238, 249, 236, 223; 713/201; 370/379, 121, 389, 399, 395, 352, 160, 54, 701, 85, 401, 400, 237, 258, 238; 711/3; 716/8**

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,262,906 A * 11/1993 Mazzola 370/54
 5,309,437 A * 5/1994 Perlman et al. 370/85
 5,550,816 A * 8/1996 Hardwick et al. 370/60
 5,604,680 A * 2/1997 Bamji et al. 716/8
 5,659,796 A 8/1997 Thorson et al.
 5,684,974 A 11/1997 Onodera
 5,692,193 A 11/1997 Jagannathan et al.
 5,737,333 A * 4/1998 Civanlar et al. 370/352
 5,761,477 A 6/1998 Wahbe et al.
 5,784,707 A 7/1998 Khalidi et al.
 5,832,224 A * 11/1998 Fehskens et al. 709/223
 5,892,912 A * 4/1999 Suzuki et al. 709/218
 6,035,105 A * 3/2000 McCloghrie et al. 709/236
 6,061,349 A * 5/2000 Coile et al. 370/389
 6,084,892 A * 7/2000 Benash et al. 370/701
 6,097,719 A * 8/2000 Benash et al. 370/352
 6,105,027 A * 8/2000 Schneider et al. 707/9
 6,128,665 A * 10/2000 Iturralte 709/238
 6,145,011 A 11/2000 Furukawa et al.
 6,167,052 A * 12/2000 McNeill et al. 370/399

6,172,981 B1 * 1/2001 Cox et al. 370/401
 6,219,699 B1 * 4/2001 McCloghrie et al. 709/221
 6,223,218 B1 * 4/2001 Iijima et al. 709/221
 6,226,751 B1 * 5/2001 Arrow et al. 713/201
 6,289,017 B1 * 9/2001 Shani et al. 370/395

OTHER PUBLICATIONS

Adve, V., et al., "Performance Analysis of Mesh Interconnection Networks with Deterministic Routing," IEEE Transactions on Parallel and Distributed Systems, vol. 5, No. 3, pp. 225-246 (3/94).

Almquist, P., "Towards Requirements for IP Routers," Network Working Group, RFC 1716, Downloaded from <http://andrew2.andrew.cmu.edu/rfc/rfc1716.html> (11/94).

Austin, T., et al., "Efficient Detection of All Pointer and Array Access Errors," Proceedings of the ACM Conference on Programming Language Design and Implementation, pp. 290-301 (6/94).

Bolla, R., et al., "A Neural Strategy for Optimal Multiplexing of Circuit-And-Packet-Switched Traffic," GLOBECOM '92 IEEE Communication Society, pp. 1324-1330 (12/92).

Brooks, R., et al., "An Optimizing Compiler for Lexically Scoped LISP," ACM Symposium on LISP and Functional Programming, pp. 261-275 (8/82).

(List continued on next page.)

Primary Examiner—Mark R. Powell

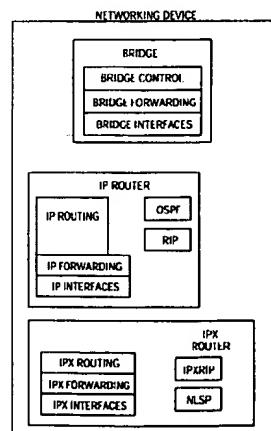
Assistant Examiner—Thong Vu

(74) *Attorney, Agent, or Firm*—Blakely, Sokoloff, Taylor & Zafman LLP

(57) **ABSTRACT**

A computer implemented method in which Internet Protocol (IP) packets are routed within a first Internet Service Provider's (ISP's) domain from a single network device with a first database. The first database includes addresses of the first ISP. IP packets are also routed within a second ISP's domain from single network device with a second database. The second database, which is separate from the first database, includes addresses of the second ISP.

81 Claims, 21 Drawing Sheets



OTHER PUBLICATIONS

Buzen, J.P., et al., "The Evolution of Virtual Machine Architecture," AFIPS National Computer Conference, pp. 291-299 (6/73).

Comer, D., *Computer Networks and Internets*, Prentice Hall (1997).

Cmelik, B., et al., "Shade: A Fast Instruction-Set Simulator for Execution Profiling," ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems, pp. 128-137 (5/94).

Dally, W.J., et al., "Deadlock-Free Message Routing in Multiprocessor Interconnection Networks," IEEE Transactions on Computers, vol. C-36, No. 5, pp. 547-553 (5/87).

Dally, W.J., "Virtual-Channel Flow Control," Proc. 17th International Symposium on Computer Architecture, pp. 60-68 (5/90).

Dally, W.J., "Performance Analysis of k-ary n-cube Interconnection Networks," IEEE Transactions on Computers, vol. 39, No. 6, pp. 775-785 (6/90).

Dally, W.J., et al., "Deadlock-Free Adaptive Routing in Multicomputer Networks Using Virtual Channels," IEEE Transactions on Parallel and Distributed Systems, vol. 4 No. 4 (466-475 (4/93).

Davidson, J., et al., "Cint: A RISC Interpreter for the C Programming Language," Proceedings of the SIGPLAN '87 Symposium on Interpreters and Interpretive Techniques, pp. 189-197 (6/87).

Deutsch, L.P., et al., "Efficient Implementation of the Smalltalk-80 System," Proceedings of the 11th Annual ACM Symposium on Principles of Programming Languages, pp. 297-302 (1/84).

Duato, J., "A New Theory of Deadlock-Free Adaptive Routing in Wormhole Networks," IEEE Transactions on Parallel and Distributed Systems, vol. 4, No. 12, pp. 1320-1331 (12/93).

Ellis, J., et al., "Safe, Efficient Garbage Collection for C++," Proceedings of the 1994 USENIX C++ Conference, pp. 143-177 (4/94).

Fischer, C., et al., "The Implementation of Run-Time Diagnostics in Pascal," IEEE Transactions on Software Engineering, vol. SE-6, No. 4, pp. 313-319 (7/80).

Gallager, R., "Scale Factors for Distributed Routing Algorithms," NTC '77 Conference Record (12/77).

Glass, C.J., et al., "The Turn Model for Adaptive Routing," Proc. 19th International Symposium on Computer Architecture, pp. 278-287 (5/92).

Goldberg, R., "Survey of Virtual Machine Research," Honeywell Information Systems and Harvard University, pp. 34-45 (6/74).

Jesshope, C.R., et al., "High Performance Communications in Processor Networks," Proc. 16th International Symposium on Computer Architecture, pp. 150-157 (5/89).

Kirkpatrick, S., et al., "Optimization by Simulated Annealing," *Science*, vol. 220, No. 4598, pp. 671-680 (5/83).

Leinwand, A., et al., *Cisco Router Configuration*, MacMillan Technical Publishing, Indianapolis, IN (1998).

Li, K., et al., "Memory Coherence in Shared Virtual Memory Systems," ACM Transactions on Computer Systems, vol. 7, No. 4, pp. 321-359 (11/89).

Linder, D.H., et al., "An Adaptive and Fault Tolerant Wormhole Routing Strategy for k-ary n-cubes," IEEE Transactions on Computers, vol. 40, No. 1, pp. 2-12 (1/91).

Luckham, D., et al., "Verification of Array, Record, and Pointer Operations in Pascal," ACM Transactions on Programming Languages and Systems, vol. 1, No. 2, pp. 226-244 (10/79).

May, C., "MIMIC: A Fast System/370 Simulator," Proceedings of the SIGPLAN '87 Symposium on Interpreters and Interpretive Techniques, pp. 1-13 (6/87).

Sites, R., et al., "Binary Translation," Communications of the ACM, vol. 36 No. 2, pp. 69-81, (2/93).

Stallings, W., *High Speed Networks: TCP/IP and ATM Design Principles*, Prentice Hall (1998).

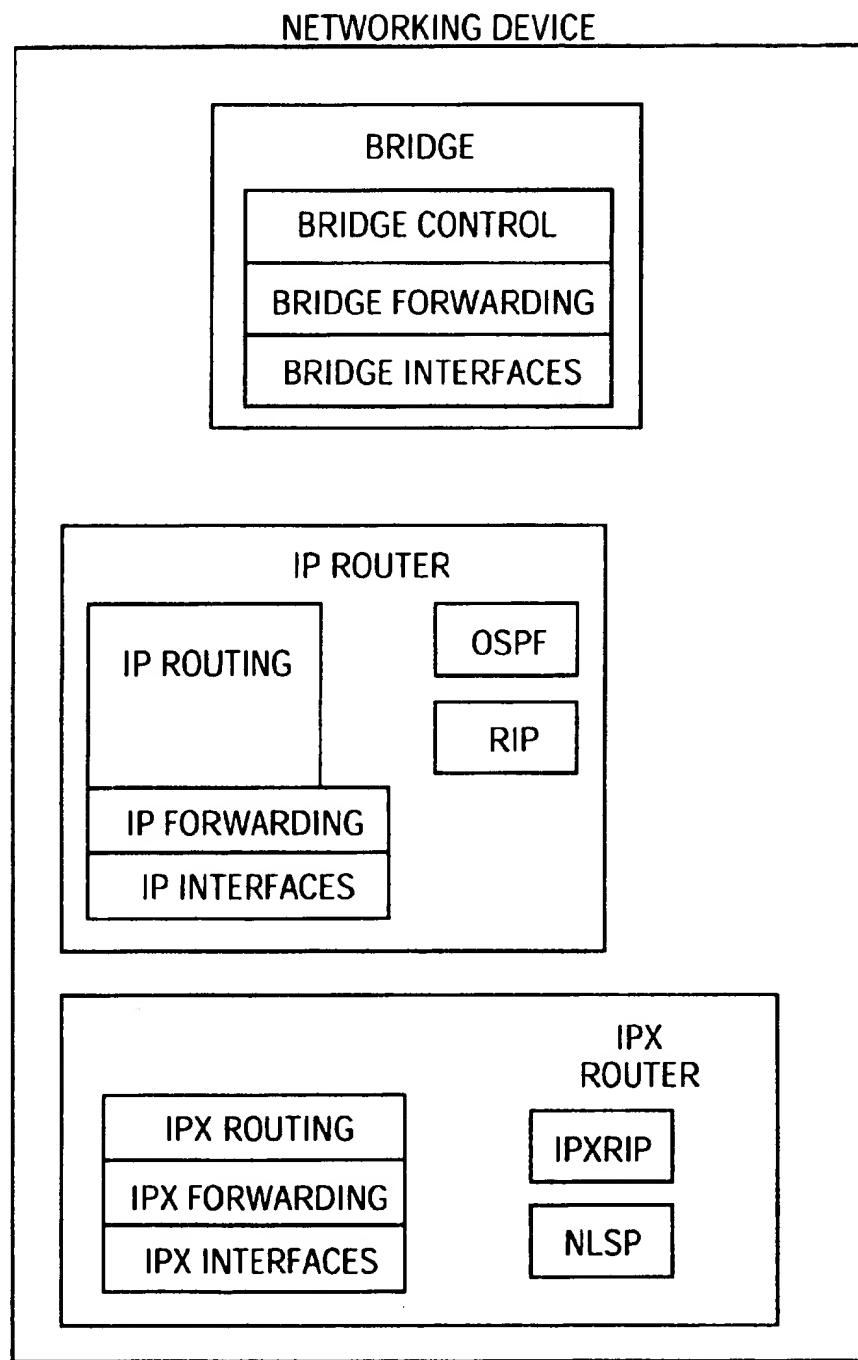
Talia, D., "Message-Routing Systems for Transputer-Based Multicomputers," IEEE Micro, No. 3, pp. 62-72 (6/93).

Wahbe, R., et al., "Efficient Software-Based Fault Isolation," Proceedings of the Symposium 14th ACM on Operating System Principles, 203-216 (12/93).

Yang, C.S., et al., "Performance Evaluation of Multicast Wormhole Routing in 2D-Torus Multicomputers," ICCI '92 IEEE Computer Society, pp. 173-178 (5/92).

Yantchev, J., "Adaptive, Low Latency Deadlock-Free Packet Routing for Networks of Processors," IEEE Proceedings, vol. 136, Pt. E, No. 3, pp. 178-186 (5/89).

* cited by examiner

**FIG. 1A**

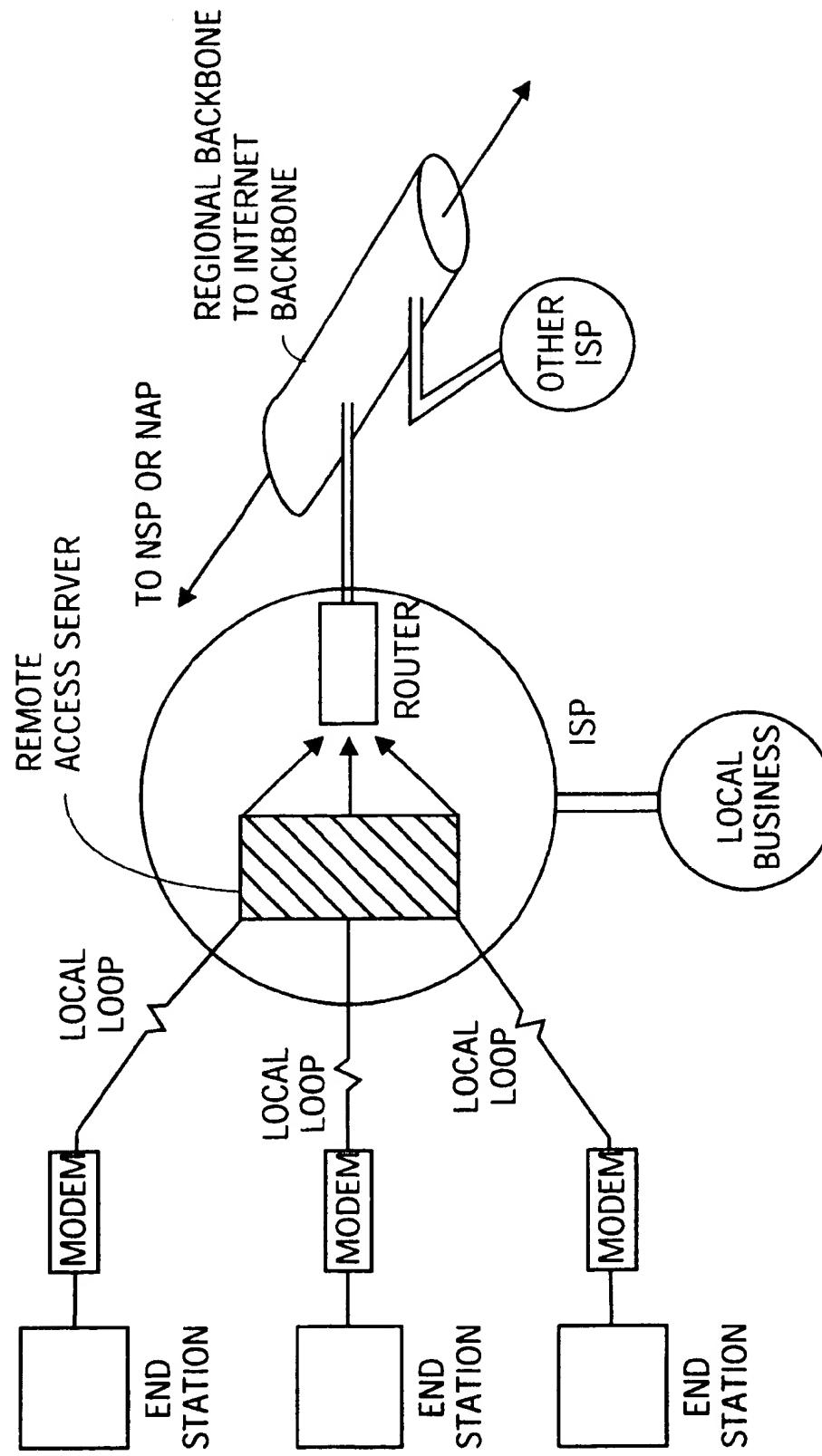
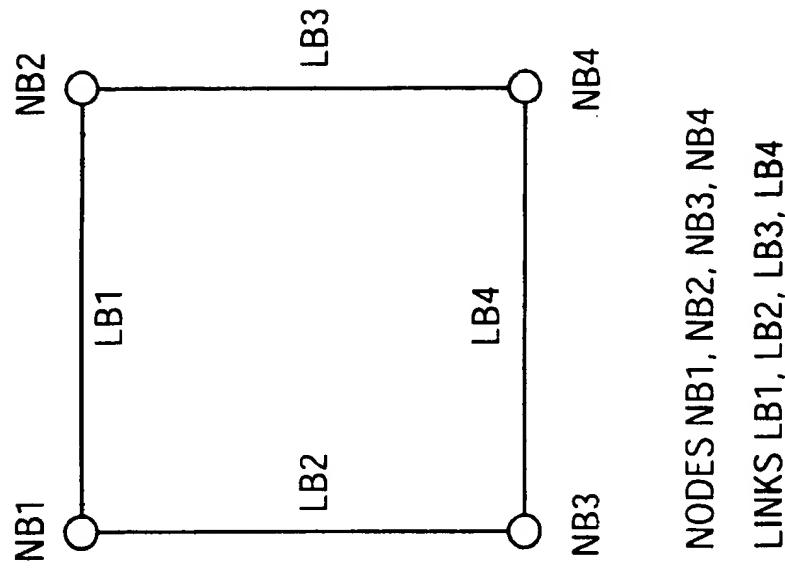
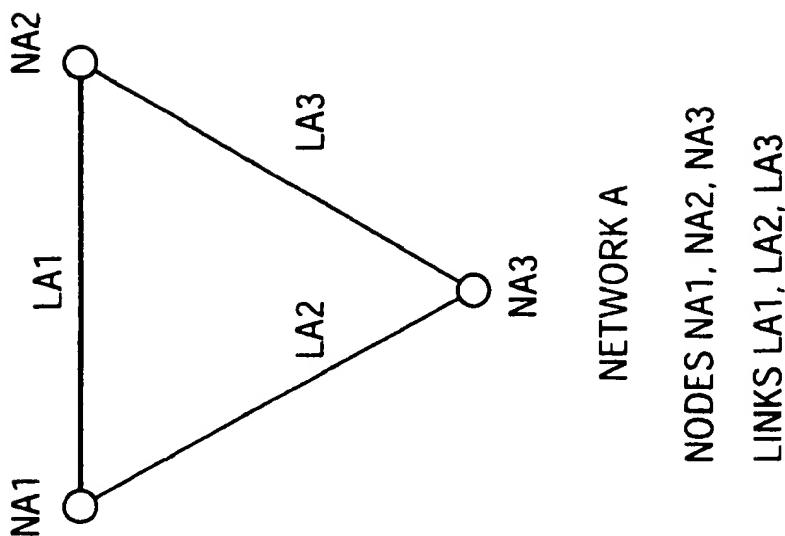
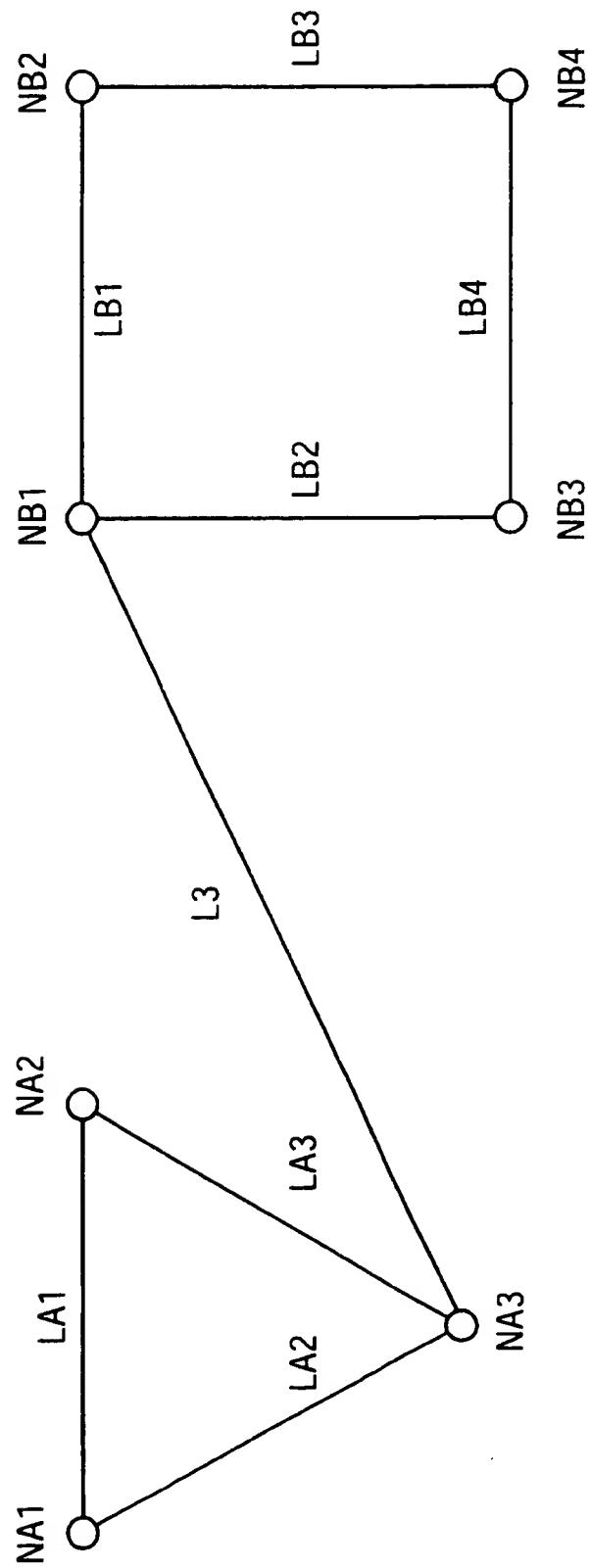


FIG. 1B

**FIG. 2B****NETWORK A**

NODES NB1, NB2, NB3, NB4
LINKS LB1, LB2, LB3, LB4

FIG. 2A

**FIG. 3**

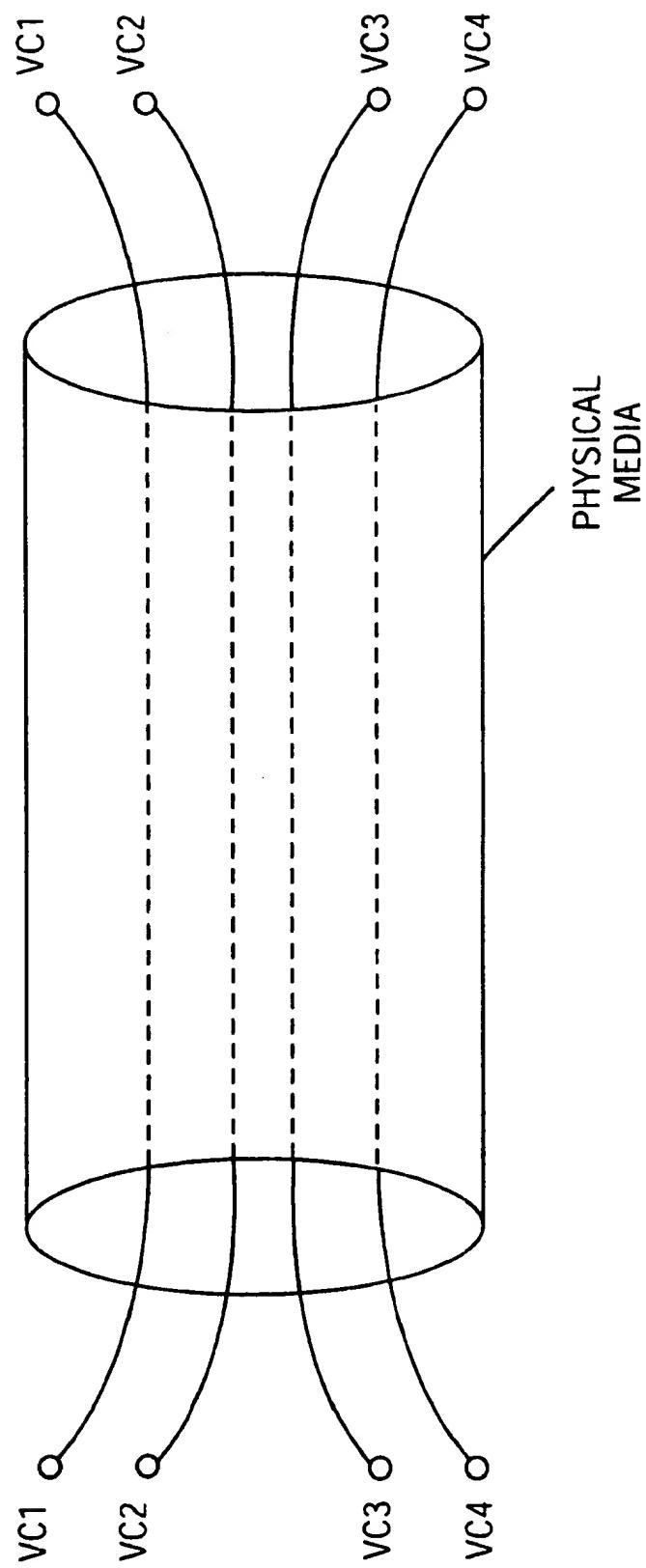
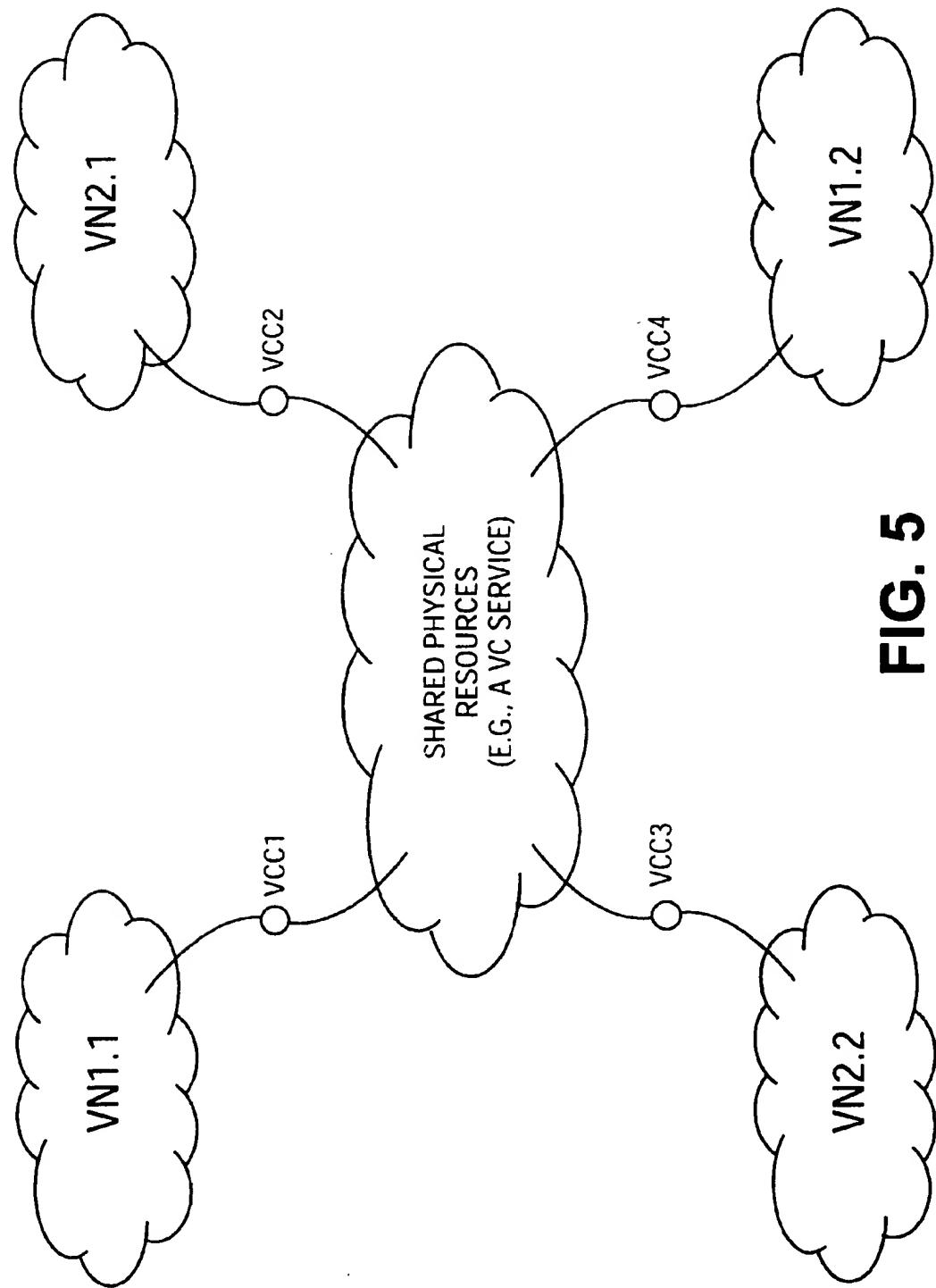
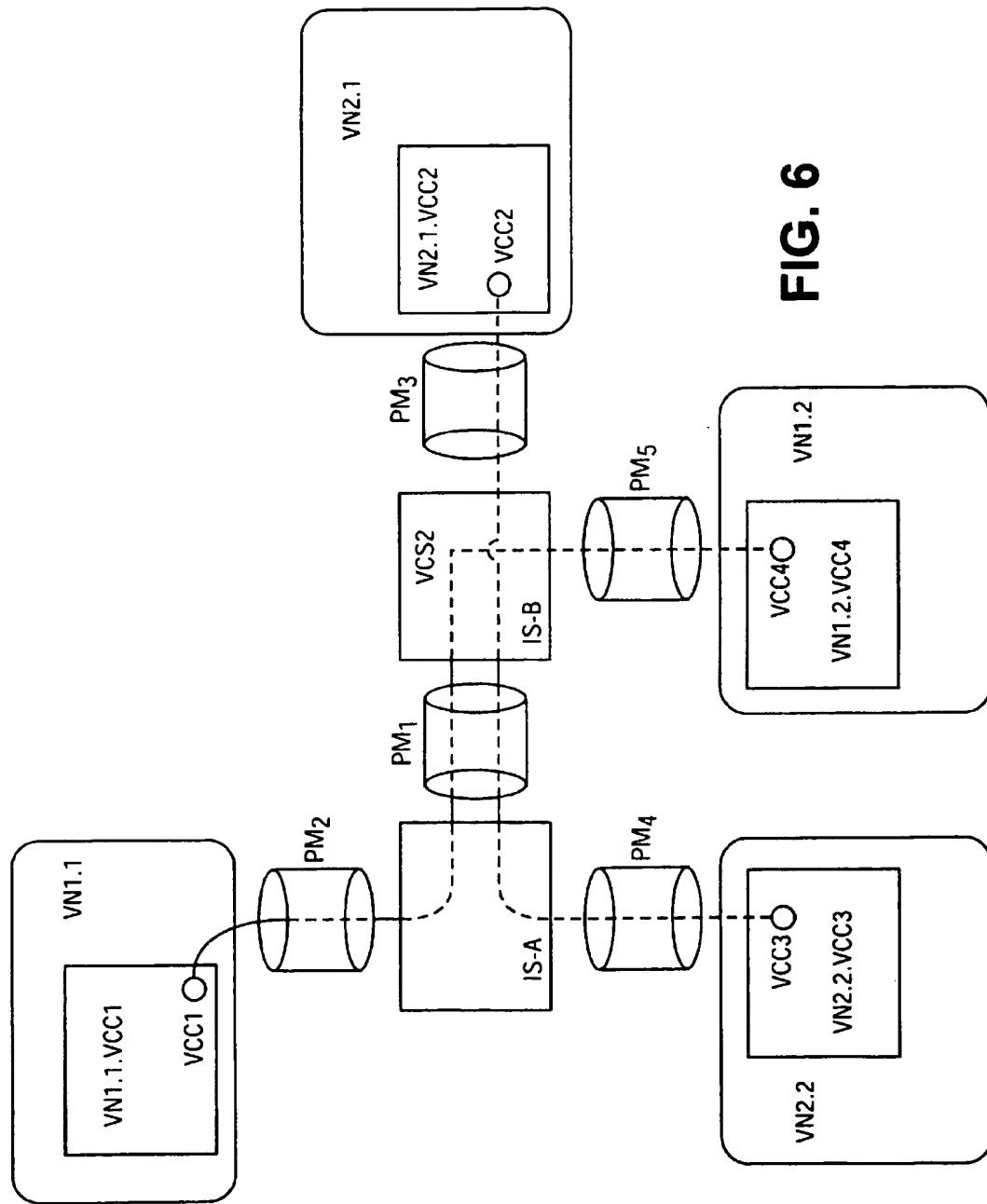


FIG. 4

**FIG. 5**



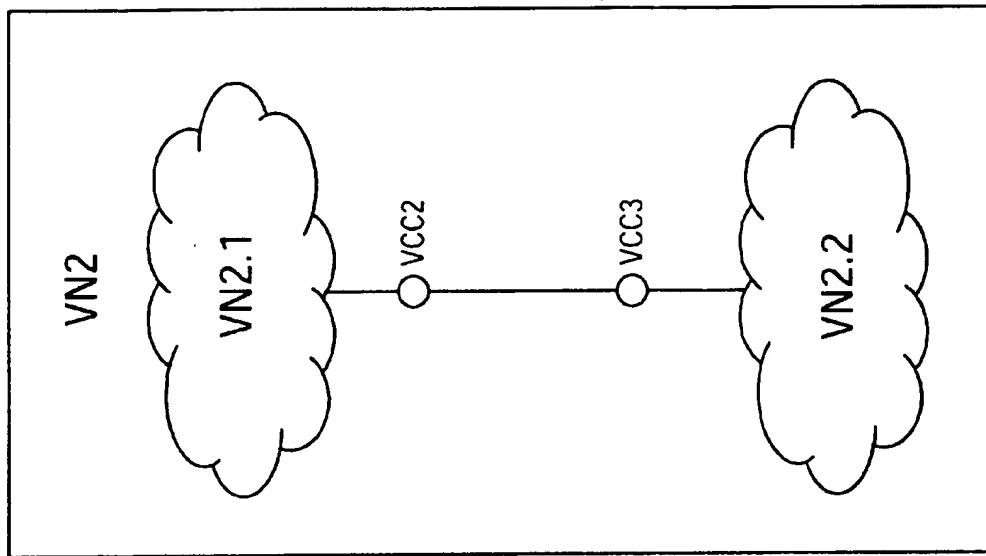
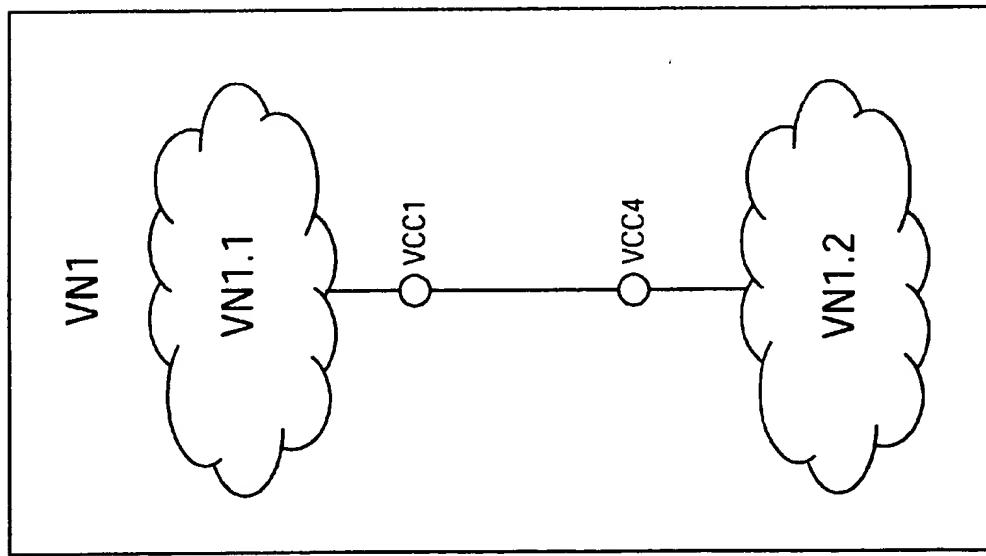


FIG. 7



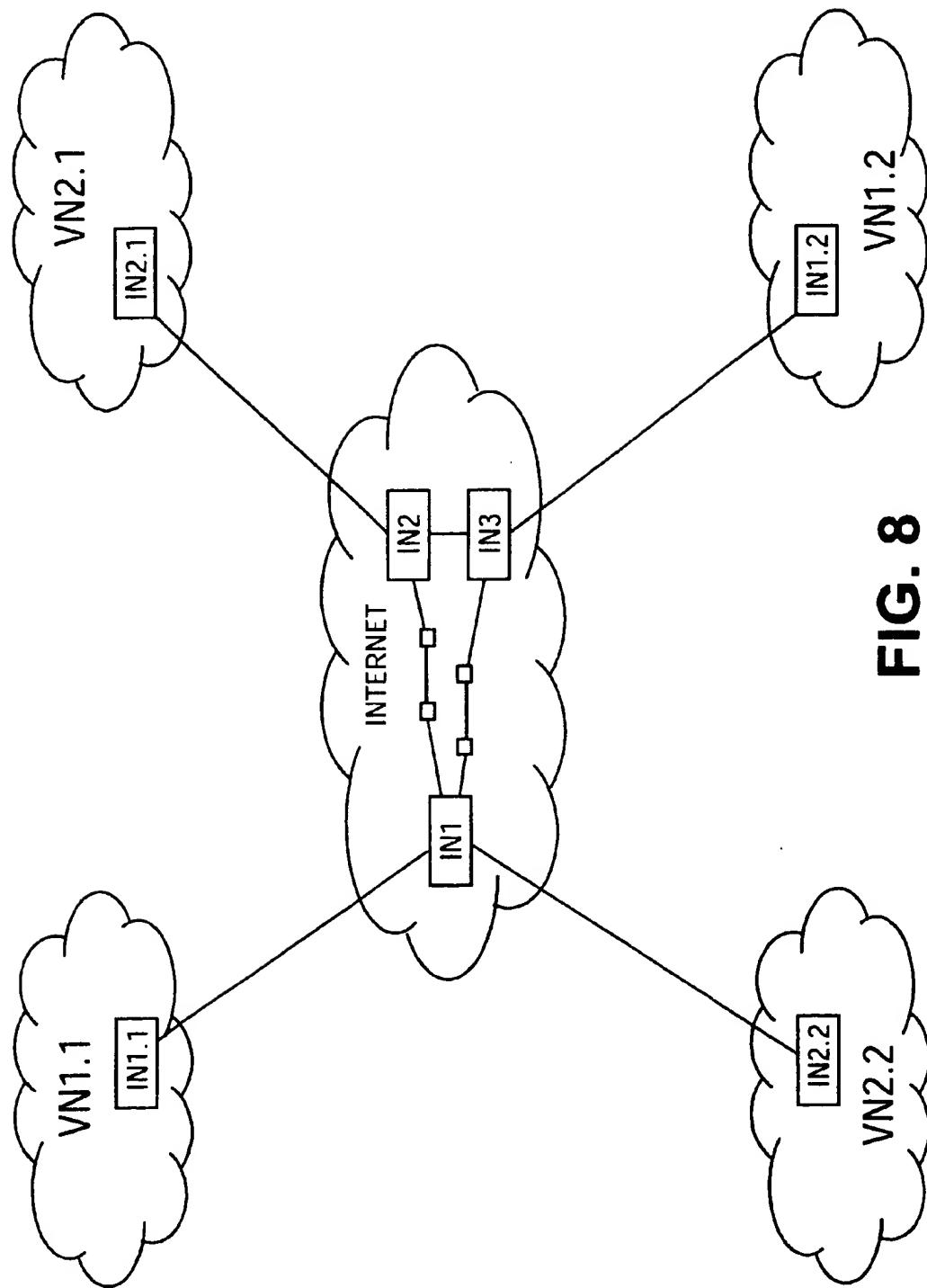
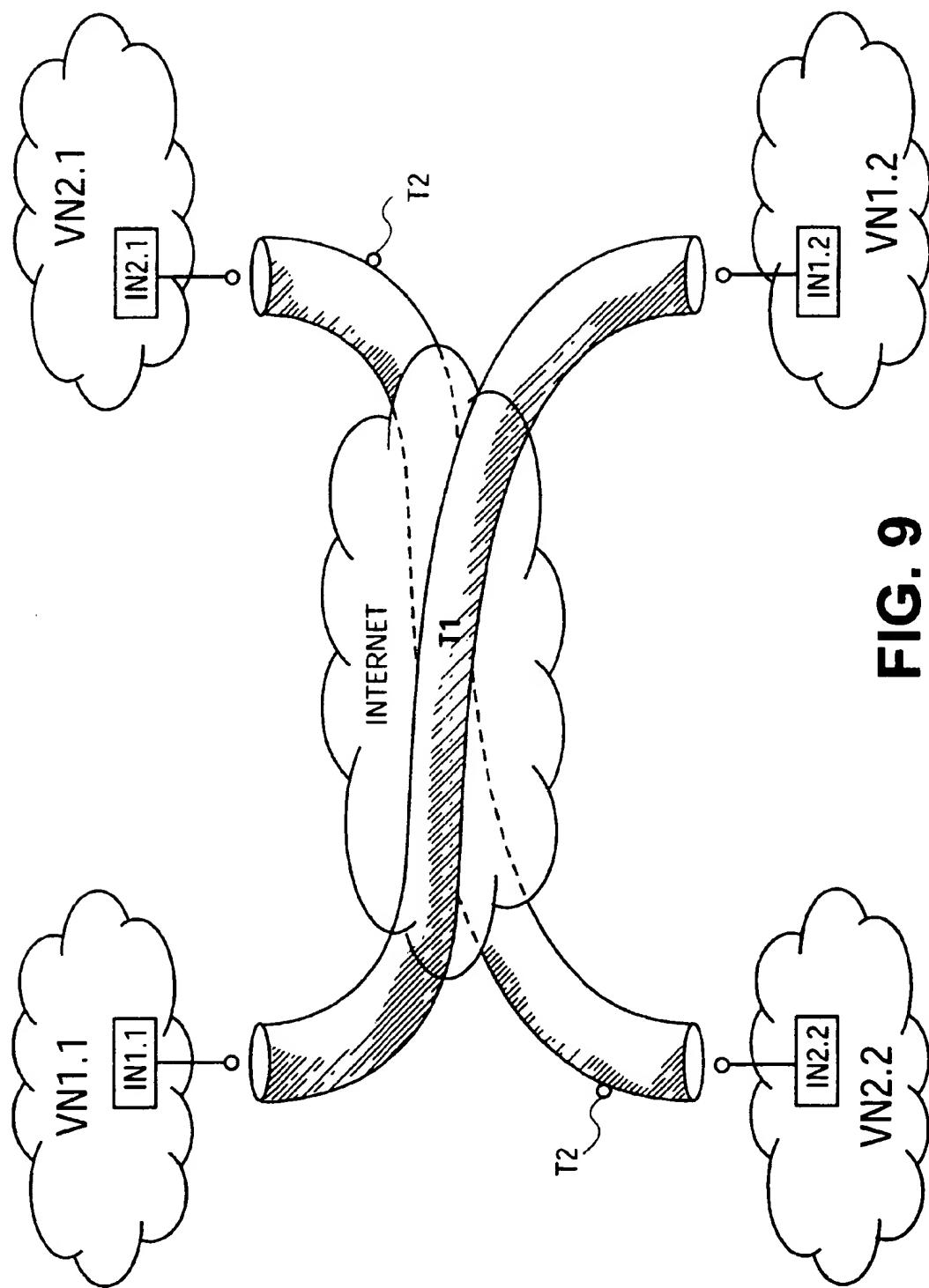


FIG. 8



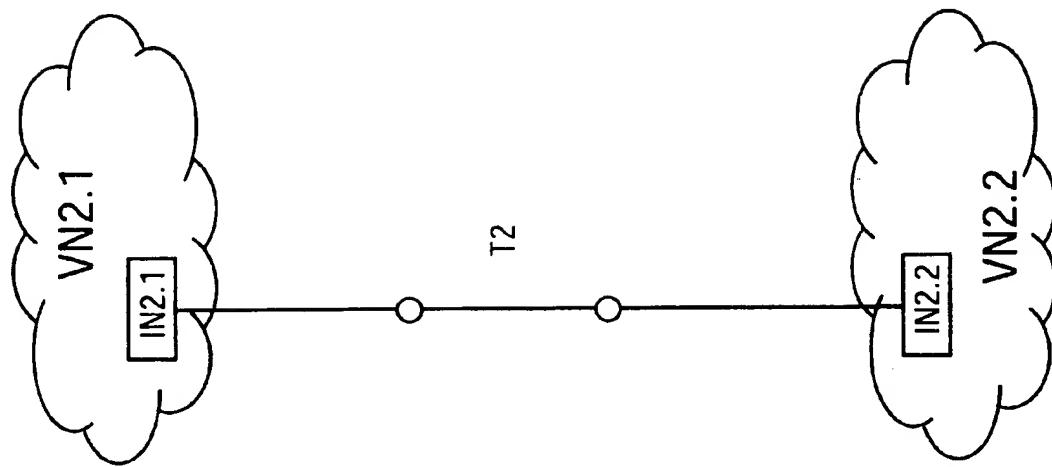


FIG. 10

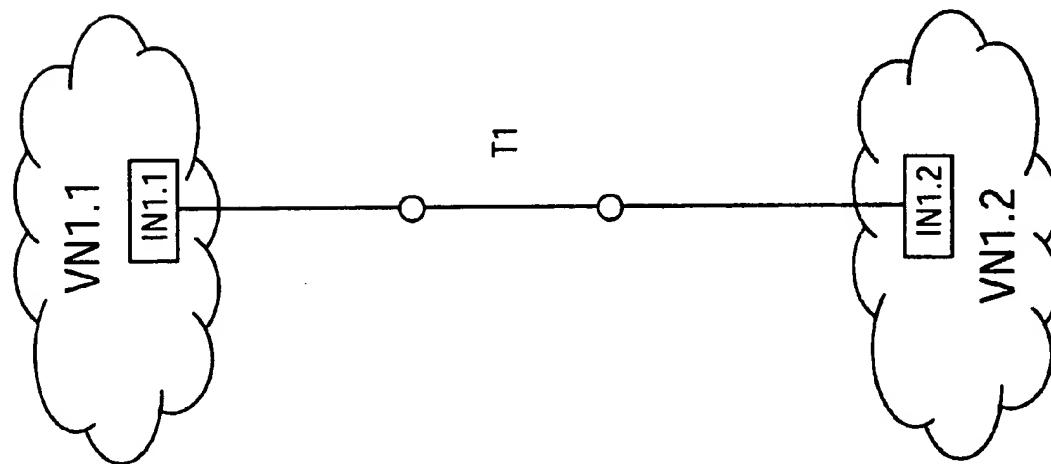


FIG. 12

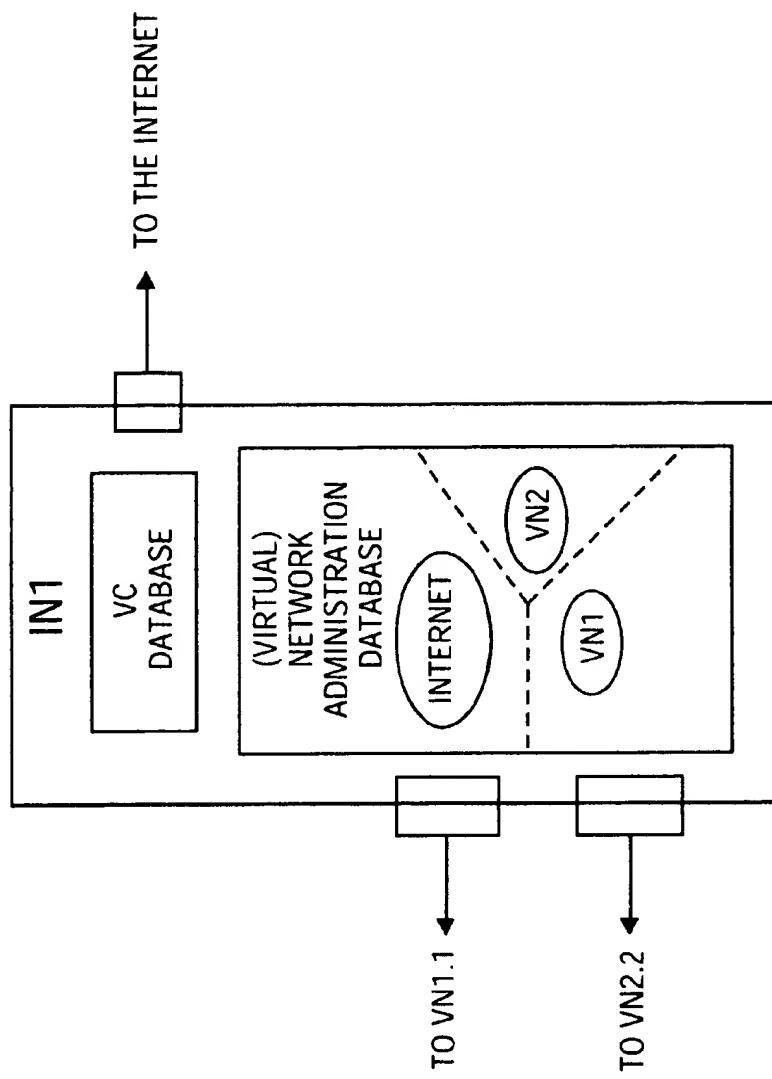


FIG. 11

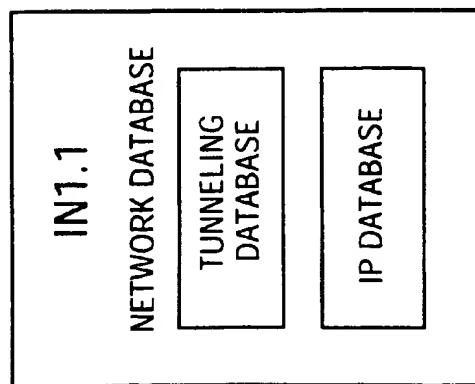
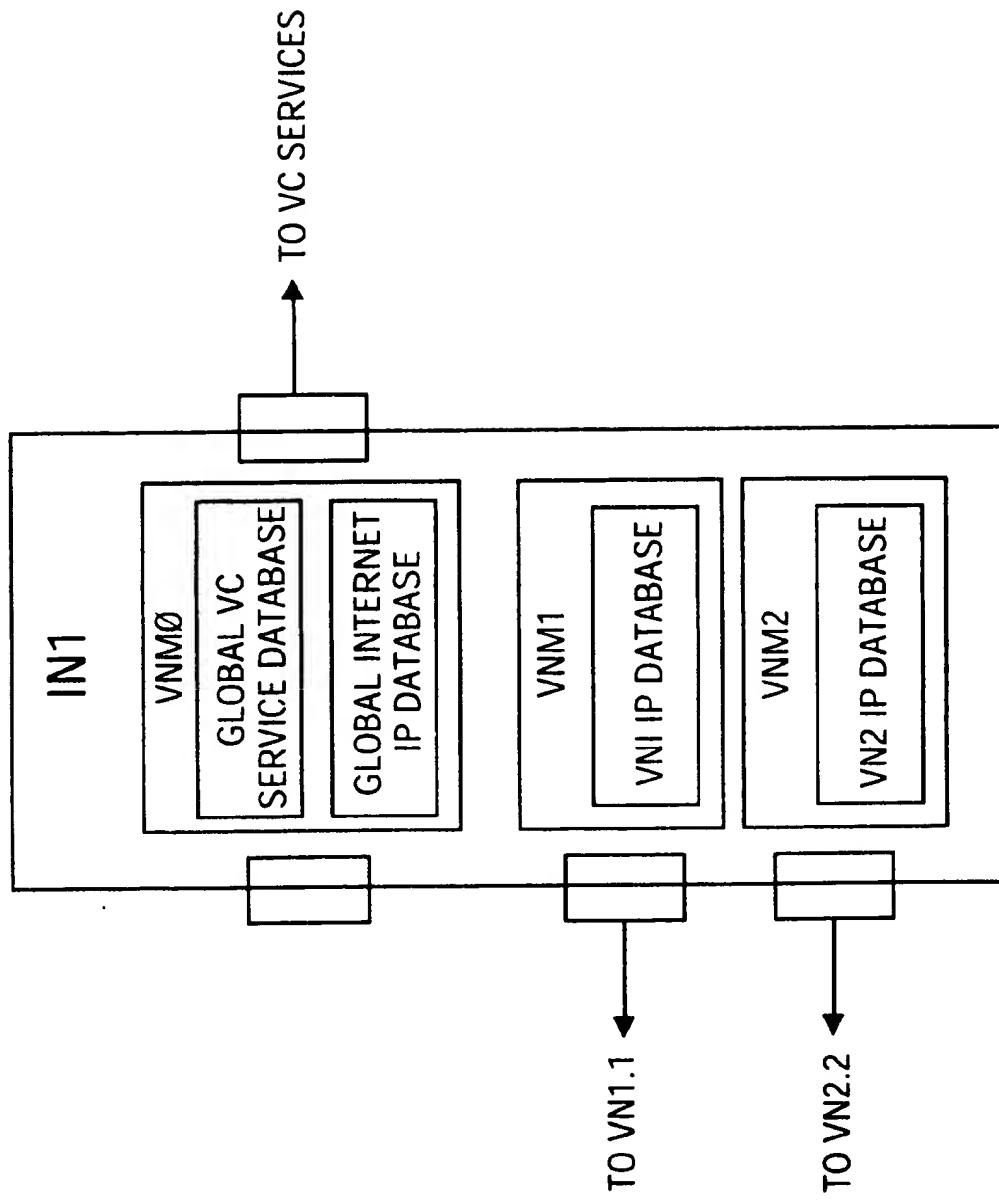
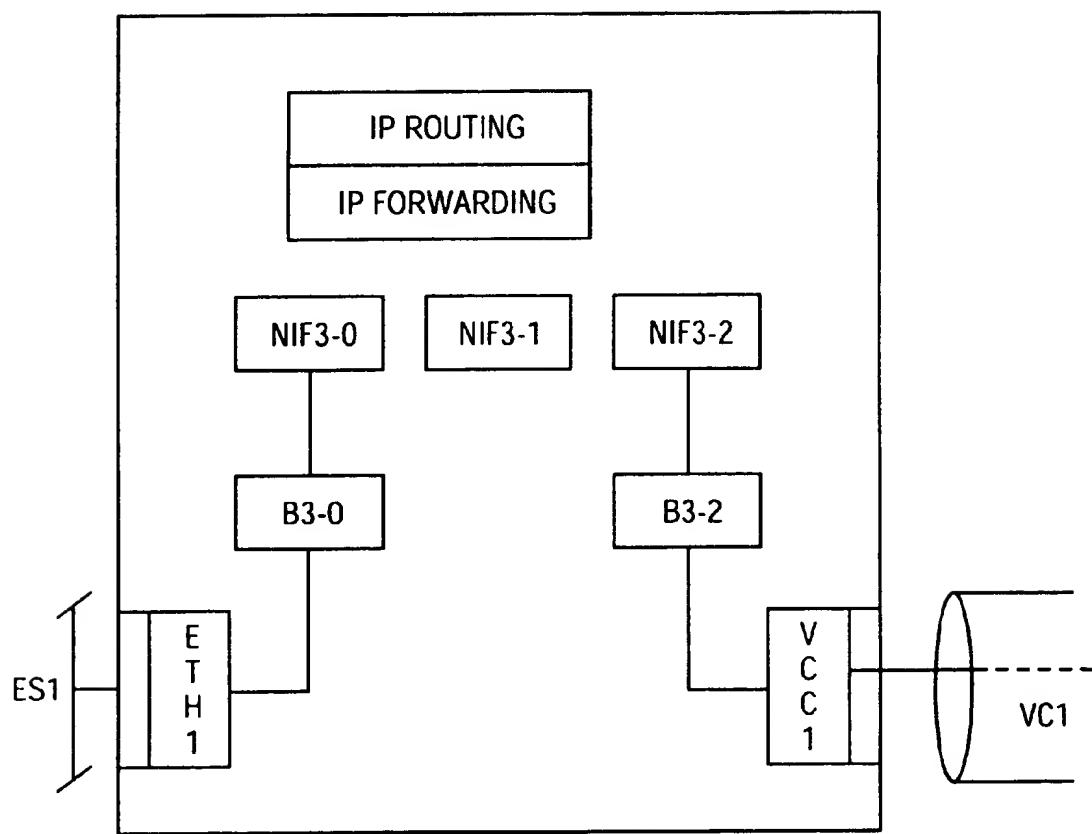


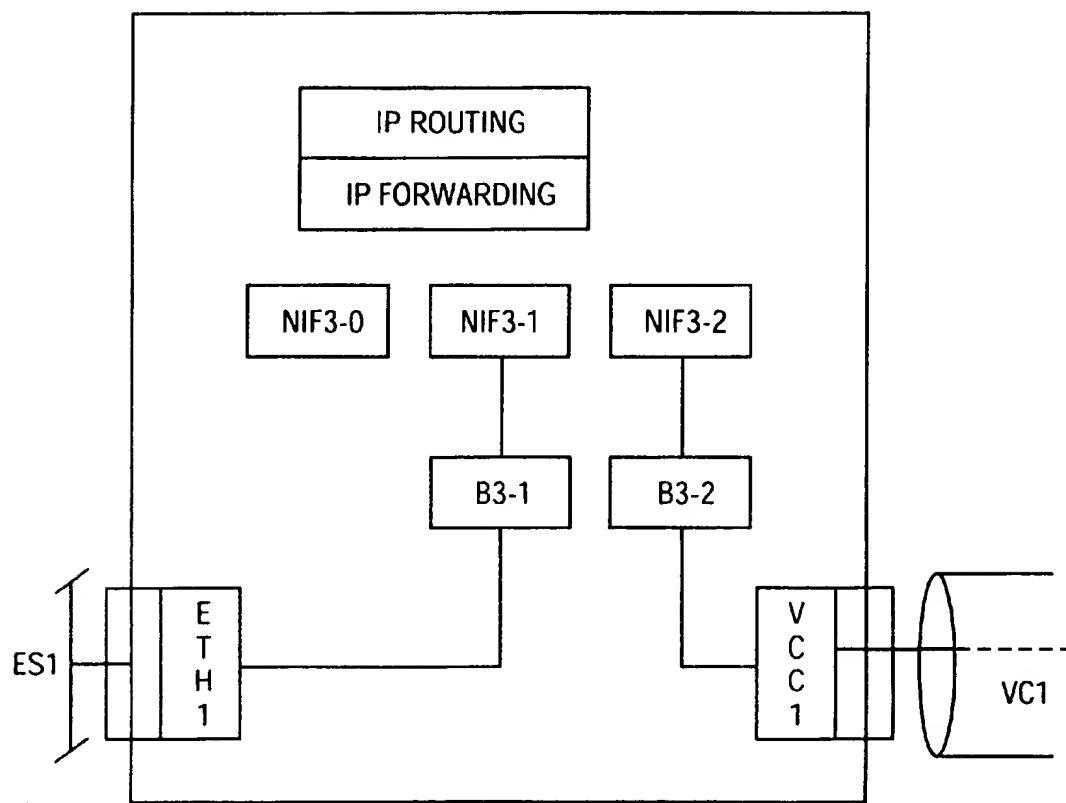
FIG. 13



IP NETWORKING DEVICE

**FIG. 14**

IP NETWORKING DEVICE

**FIG. 15**

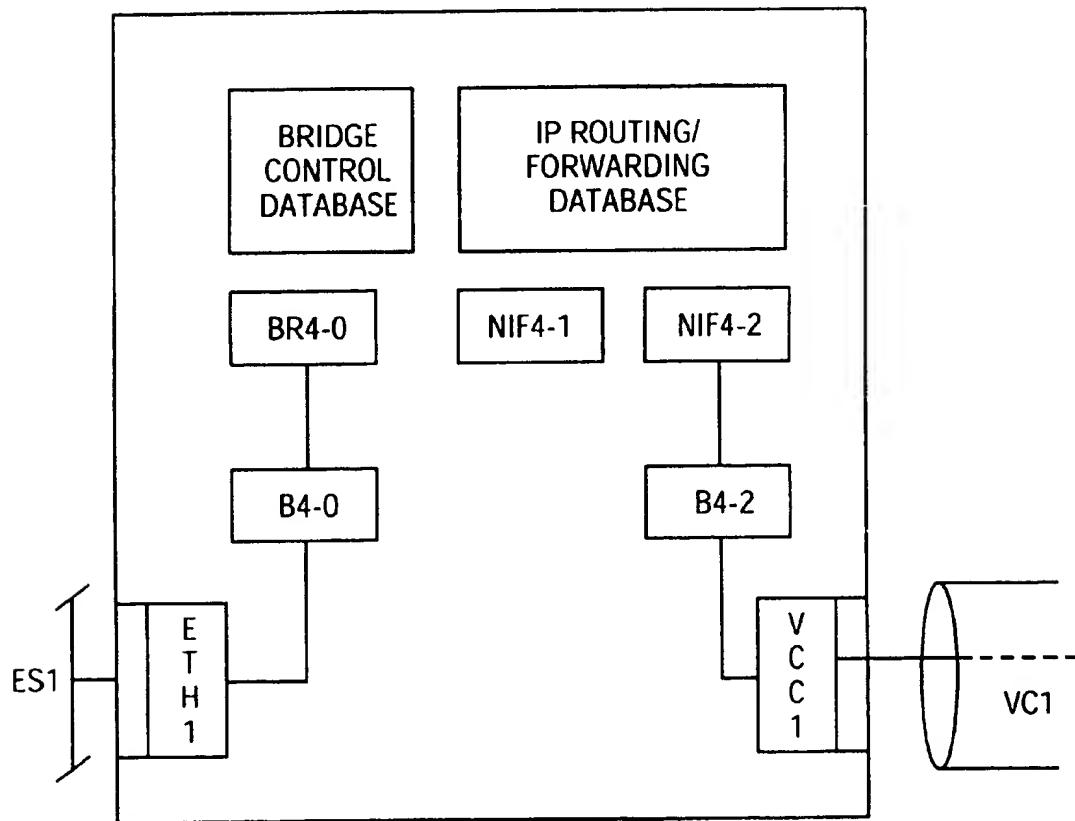
BRIDGING AND IP ROUTING
NETWORK DEVICE

FIG. 16

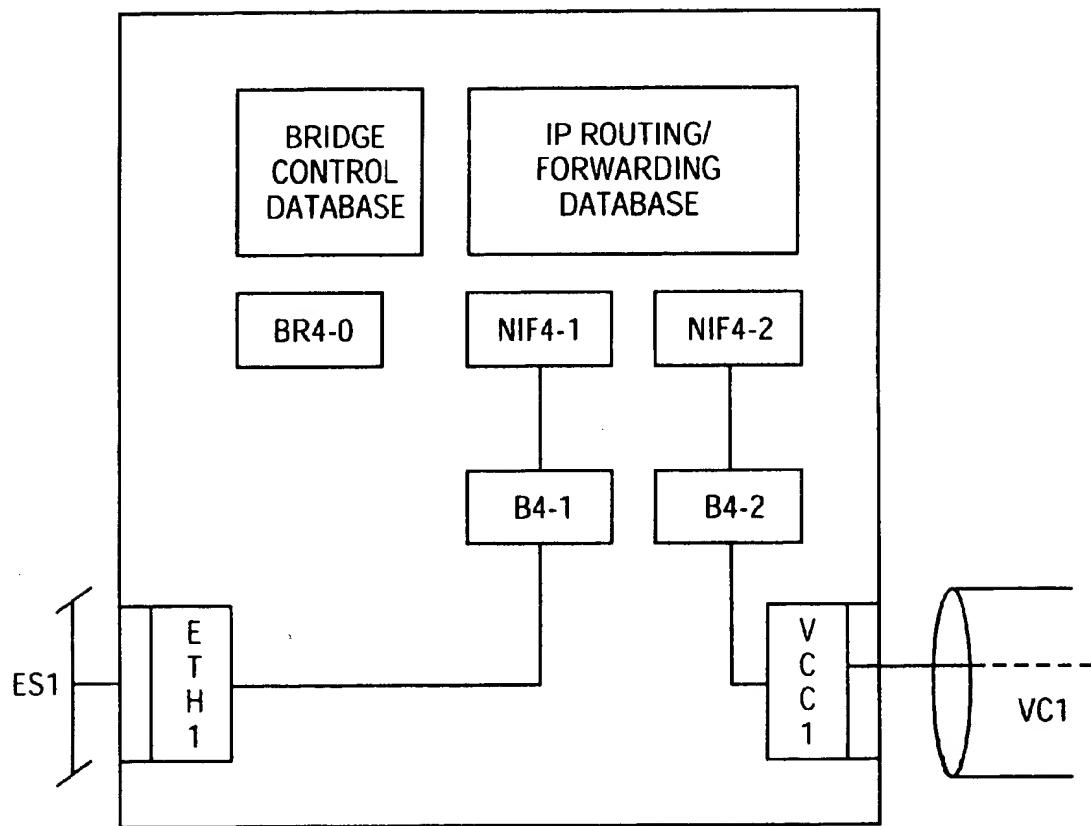
BRIDGING AND IP ROUTING
NETWORK DEVICE

FIG. 17

IP ROUTER NETWORK DEVICE

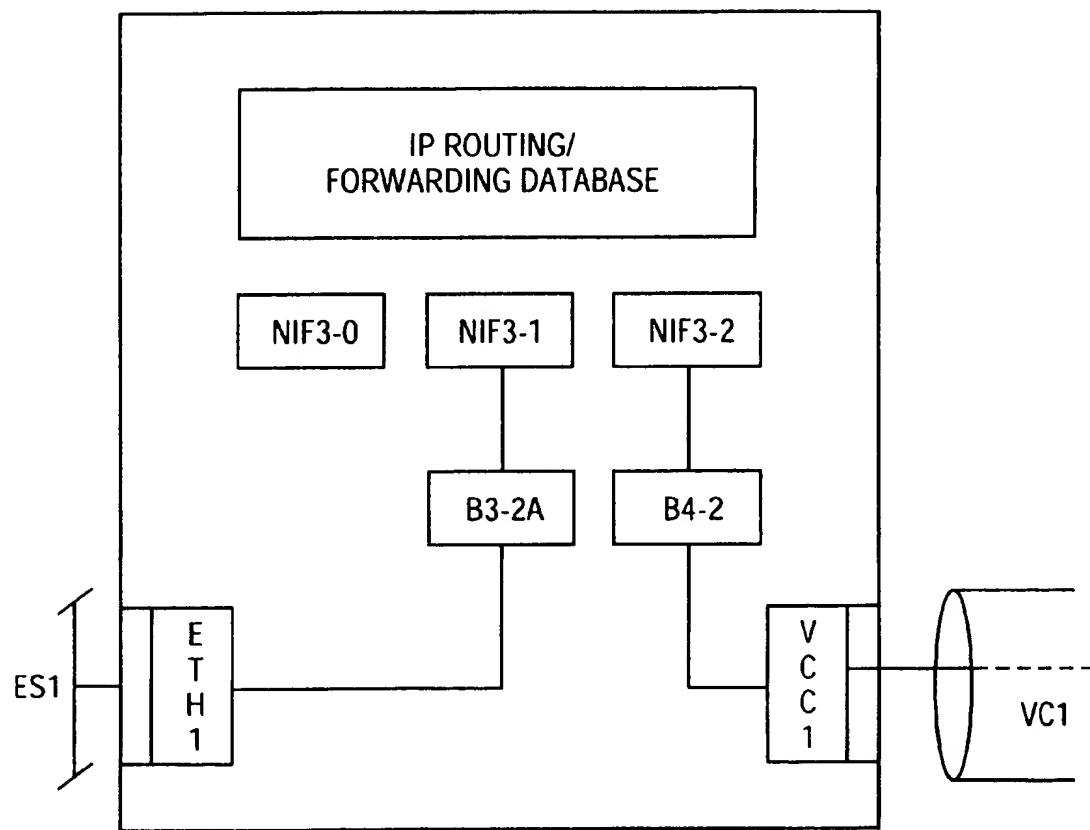


FIG. 18

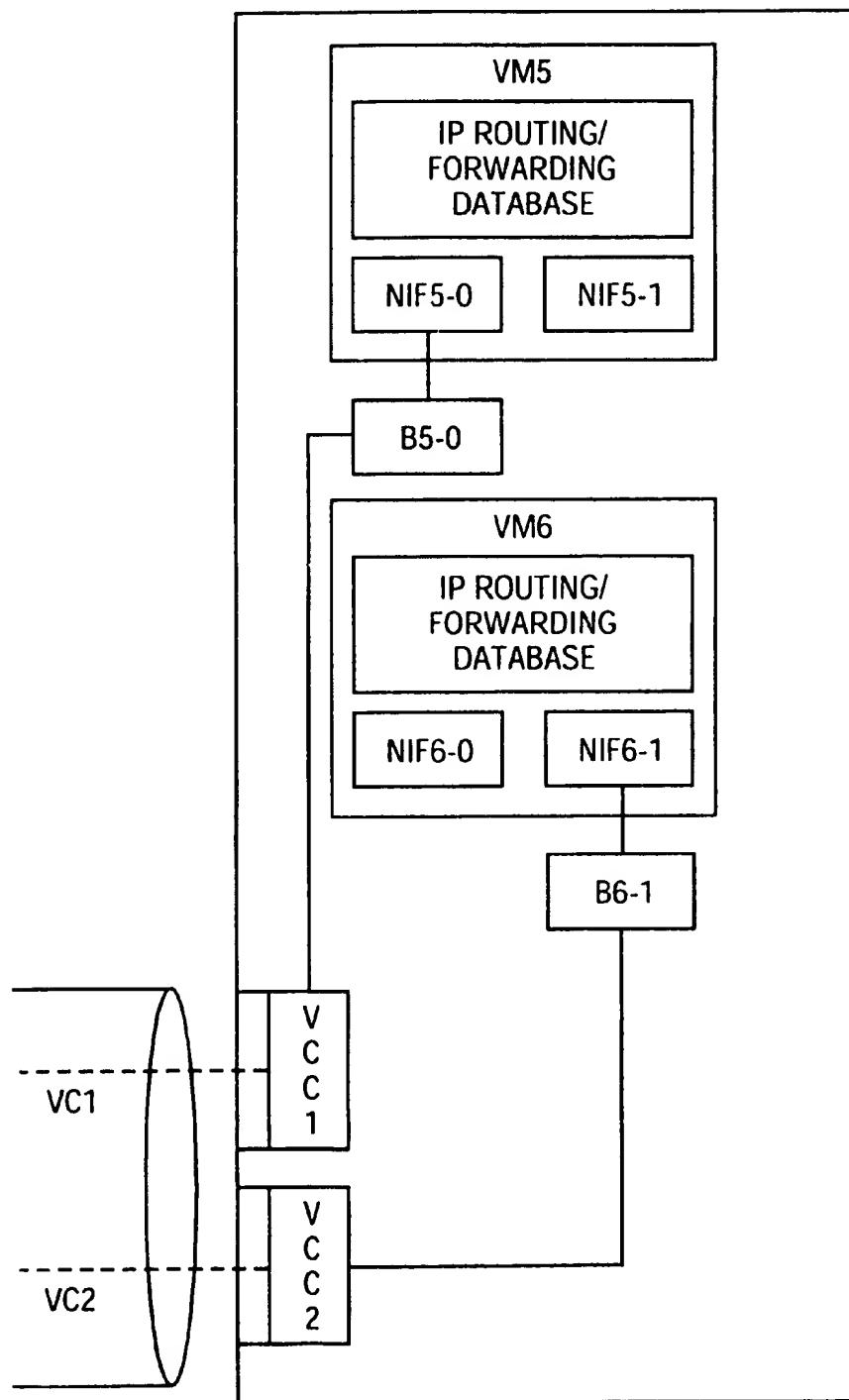
MULTIPLE IP ROUTERS IN A
SINGLE NETWORK DEVICE

FIG. 19

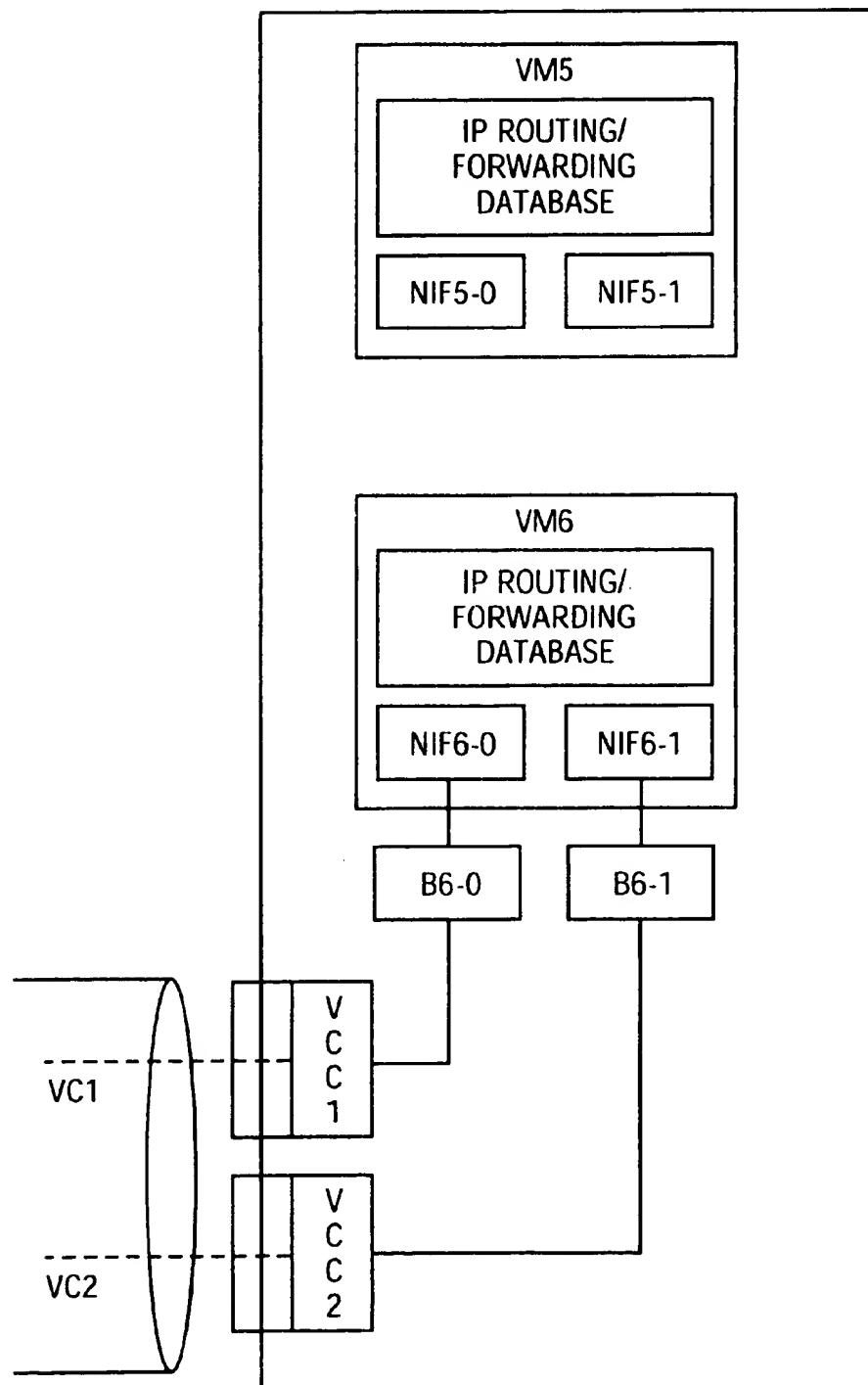
MULTIPLE IP ROUTERS IN A
SINGLE NETWORK DEVICE

FIG. 20

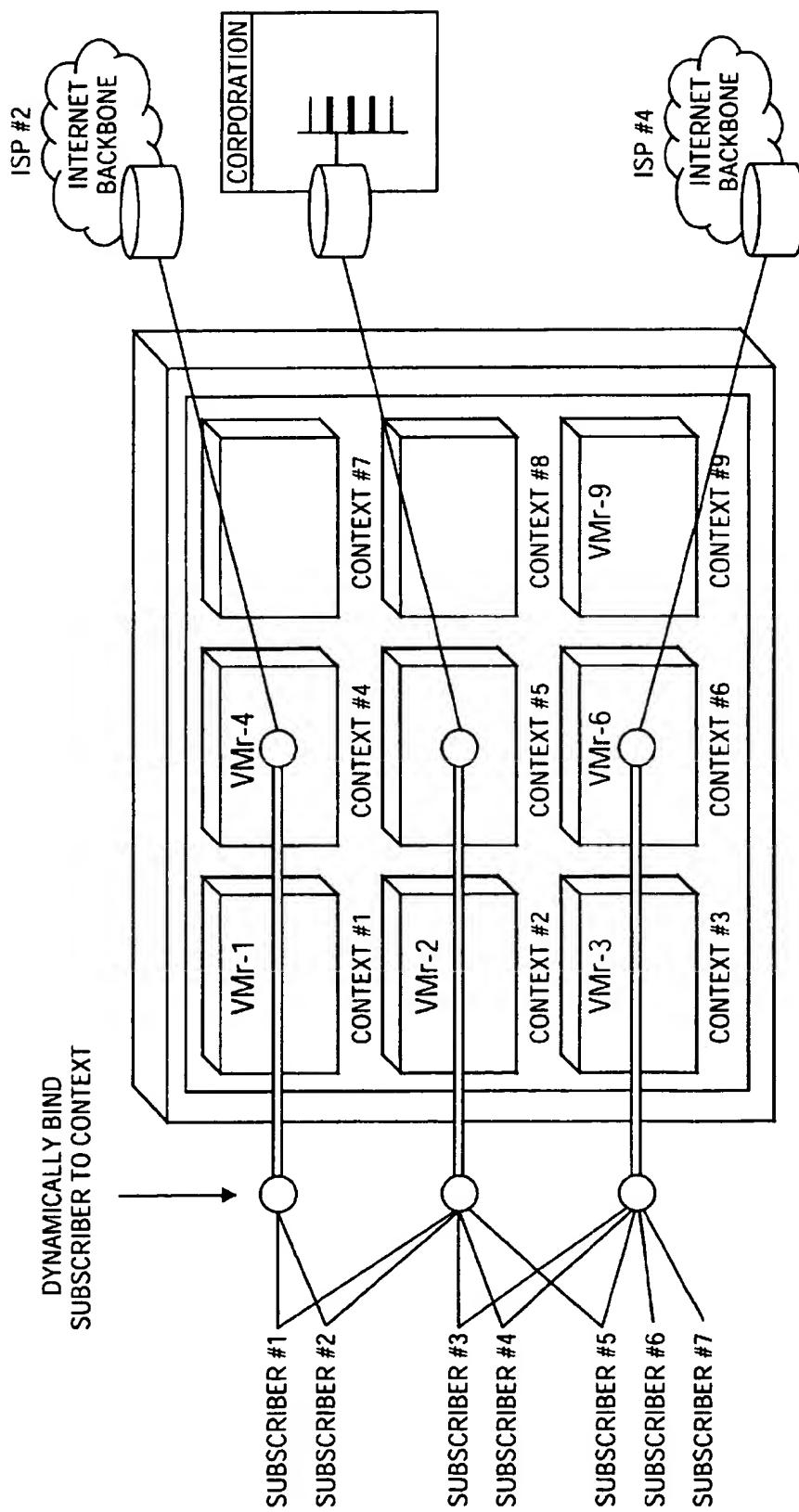


FIG. 21

DOMAIN ISOLATION THROUGH VIRTUAL NETWORK MACHINES

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates in general to communications networks, and more particularly, to the operation of network devices that can operate in multiple virtual networks simultaneously.

2. Description of the Related Art

Network Layering and Protocols

A communication network provides information resources transfer services that transfer information resources among devices attached to the network. Information resources, as the term is used herein, includes any form of information that can be transmitted over a network for use by or with any end station or network device connected to the network. Information resources, for example, may include computer programs, program files, web pages, data, database information, objects, data structures, program icons, graphics video information or audio information. *Computer Networks and Internets*, Douglas E. Comer, Prentice Hall, 1997, provides extensive information about communication networks.

Networks are built from devices or stations called nodes, and the communications channels that interconnect the nodes, called links. A set of nodes and links under one administrative authority is called a network domain. Communication between end stations attached to a network ordinarily is achieved through the use of a set of layered protocols. These protocols are generally described by reference to the Open Systems Interconnection (OSI) computer communications architecture. The standard OSI architecture includes seven layers: application, presentation, session, transport, network, data link and physical. A communication network may employ fewer than the full seven layers. However, the layer 2 and the layer 3 software protocols ordinarily play a prominent role in the transfer of information between interconnected networks and between end stations connected to the networks.

The physical layer is the lowest layer (layer 1) of the OSI model. There are numerous technologies that can be employed to build networks at layer 2. Layer 2 networks can be "connection oriented", meaning that a connection must be established before data can flow between two stations; ATM, Frame Relay, and X.25 are examples of connection oriented layer 2 protocols. Layer 2 networks can also be connection-less, meaning data can be transmitted without establishing any connection in advance; Ethernet and FDDI are two examples of connection-less layer 2 protocols.

In order to provide services useful to end users, the devices in a network must perform higher layer functions to create what are called "virtual networks". The "Internet" is one example of a very popular and public virtual network. The Internet uses the IP protocol to provide the higher layer (layer 3) functions required to support operation of the virtual network. There are many other private (virtual) networks that also uses the IP protocol. The term "internet" with a small "i" is used to differentiate between these less well known private internets, and the very popular and public large "I" Internet. There are many other protocols that can be used to construct virtual networks at layer 3, including IPX, DECnet, AppleTalk, CLNP, etc. There are many

other private and public networks using these other layer 3 protocols, either independent of or in conjunction with the IP protocol.

Thus, networks can be built at many different layers. Each layer has its own function and its own type of nodes and links. Higher layer networks are built "on top of" lower layer networks. In other words, nodes at a given layer may use the services of the next lower layer to provide links for communication with peer nodes (i.e. nodes at the same layer on other devices). Routers are examples of nodes in a layer 3 network. Bridges are examples of nodes in layer 2 networks.

Network Domains

A network domain as the term is used herein refers to the set of nodes and links that are subject to the same administrative authority. A single administrative authority may administer several networks in separate domains, or several layers of the same network in a single domain, or any combination. There are actually several possible administrative domains in any large virtual network. The boundaries of a network domain can be defined along the lines dividing layers of the protocol stacks. For instance, the same layer 1 physical devices and physical connections may have several layer 2 network domains layered onto them. These layer 2 domains, in turn, may have one or more layer 3 domains layered on top of them. A network domain may even transcend the boundaries between layers such that a layer 2 network and a layer 3 network may be part of the same network domain.

The administration of even a single network domain can be quite complex. Virtual networks have administrative authorities associated with them to control their higher layer functions. The cost of administering a network, physical or virtual, can be enormous, and is often the largest cost item in the operations of a network.

When several virtual networks are layered on top of the same layer 2 service or another virtual network, the boundaries between network domains may be somewhat obscure. The boundaries between the domains of the overlaid virtual networks intersect at points where they must share physical or virtual resources. In practice, the administrators of the overlaid virtual networks are very concerned about sharing resources, especially when they are competing commercial entities. Concerns arise about integrity, privacy, and security of data and network control information flowing across the shared resources at the lower layers. The administrators of the underlying networks are called upon to solve complex administrative problems. The costs of administering these networks increases quickly with the number of virtual networks, their size, the complexity and compatibility of their individual policies, and increased demands for security, integrity, and isolation between domains.

Network Devices and Databases

The term network device is used here to refer to the collection of mechanisms (e.g. computer and communications hardware and software) used to implement the functions of a station in a network. A network device contains some capacity to store and operate on information in databases in addition to the ability to transmit and receive information to and from other devices on the network. Examples of network devices include but are not limited to routers, bridges, switches, and devices that perform more than one of these functions (e.g. a device that does both routing and bridging).

A router is an example of a network device that serves as an intermediate station. An intermediate station is a network

device that interconnects networks or subnetworks. A typical router comprises a computer that attaches to two or more networks and that provides communication paths and routing functions so that data can be exchanged between end stations attached to different networks. A router can route 5 packets between networks that employ different layer 2 protocols, such as Token Ring, Ethernet or FDDI, for example. Routers use layer 3 protocols to route information resources between interconnected networks. Nothing precludes a network device that operates as an intermediate station from also operating as an end station. An IP router for example typically also operates as an end station.

A router can understand layer 3 addressing information, and may implement one or more routing protocols to determine the routes that information should take. A multiprotocol 10 router runs multiple layer 3 protocols such as IP, IPX or AppleTalk for example. A router also be characterized as being multiprotocol if it runs multiple adaptive routing protocols such as RIP, BGP or OSPF all feeding a single IP layer.

The network device router configuration of FIG. 1A depicts what is often referred to in industry as a multiprotocol bridge/router. In this illustrative example, there are separate databases for three layer 2/3 networking protocols: bridging, IP routing, and IPX routing. The example IP database employs both the OSPF and RIP dynamic routing protocols. Thus, the intermediate station node of FIG. 1A includes both multiple networking protocols and multiple routing protocols.

A bridge is another example of a network device that serves as an intermediate station. A typical bridge comprises a computer used to interconnect two local area networks (LANs) that have similar layer 2 protocols. It acts as an address filter, picking up packets from one LAN that are intended for a destination on another LAN and passing those packets on. A bridge operates at layer 2 of the OSI architecture.

The term network database will be used to refer to all the control information housed in a network device required to support the device's operation in a set of one or more networks. Each device in a network holds its own network database. In order for the network at large to operate properly, the network databases of all network devices in a network domain should be consistent with each other. The network database control information defines the behavior of its network device. For example, not only might it determine whether the network device will function as a router or a bridge or a switch, but also it will determine the details of how the device will perform those functions.

When a network device is deployed to operate in multiple domains, its network database can become quite complex. The cost of administering the network device increases significantly when the network database is more complex. The cost of administration is already the most significant cost of operating many networks, and the trend toward greater complexity through greater use of virtual networking continues unabated.

The information found in a typical network database includes, but is not limited to, data used to configure, manage, and or monitor operations of:

Communications Hardware (e.g. layer 1 transceivers/ drivers/chips etc.)

Computer Hardware

Computer Software

Layer 2 Addressing

Layer 2 Connections (Layer 2 interfaces)

Traffic filter policies

Bridging (IEEE 802.1D)

Bridge filters and or policies

Network (layer 3) Addressing

Layer 3 Connections (Layer 3 interfaces)

(Network/layer 3) Address Translation (NAT) policies

Access Control (e.g. user names and password)

Access policies (e.g. what user can use what services)

Routing (IETF RFC 1812)

Routing Protocols (e.g., BGP, OSPF, RIP, IGRP, etc.)

Route filters and policies (e.g. route leaking)

Tunneling

Tunneling Protocols (e.g., L2TP, GRE, PPTP, etc.)

A single network device can operate in one or more (virtual) network domains. For each domain in which a device operates, it needs to store information about that domain in some database form.

Much of the information in a network database must be configured manually; particularly the policy information as it must reflect the administrator's subjective wishes for how the network should operate. Manual configuration involves

25 human effort, which can become expensive, especially as the number of policies and their complexity increases. Network administrative chores include the assignment of user names, passwords, network addresses or other user identifiers, and configuration of policy databases. This configuration and

30 management may be used to establish traffic filtering policies such as what kind of information payloads will be carried. Traffic and Route filtering policies may be established to determine what paths through the network will be used for each payload carried. Access control policies may be to dictate which users at which end stations have access to which services at other end stations. Security policies may be established to ensure the integrity of the information payloads. Each configured bit of policy somehow finds its way into the network database of the device implementing

40 the policy.

Cisco Router Configuration by A. Leinwand, B. Pinsky and M. Culpepper, published by MacMillan Technical Publishing, Indianapolis, Ind., 1998 provides an extensive treatment of the configuration of the databases of Cisco 45 System routers. This is just one example of a network device database.

Building Virtual Networks

The layering of software protocols in accordance with the 50 ISO architecture makes possible the creation of "virtual networks". Virtual networks are to be contrasted with physical networks. Two physical networks which have no physical devices or links in common, can be said to be physically isolated from each other. Physical isolation may be required in order to ensure that a network has the highest levels of security and integrity.

Physical networks are defined at layer 1 of the OSI model. 55 Virtual networks, on the other hand, are created at higher layers. It is possible to create multiple virtual networks all sharing common physical resources. A network is definitely virtual if it shares a common physical medium or device, such as an intermediate station, with any other (virtual) network. There are many conventional technologies and many commercially available products which can be used to 60 build many types of virtual networks. For example, virtual circuits are a layer 2 construct that can be employed to create virtual networks.

It has been common practice in the industry for phone companies to offer connection oriented layer 1 and 2 services to Internet Service Providers (ISPs), corporations, and residential customers. These customers may build one or more higher layer (layer 3 and above) virtual networks on top of such publicly available layer 1 and 2 services. The higher layer virtual networks share a common set of layer 1 and 2 services, each having its private set of virtual circuits.

A PC or a server are examples of end stations. End stations located at home or business, for example, may connect into an internet through an internet service provider (ISP). There are regional, local and global ISPs. In most cases, local ISPs connect into the regional ISPs which in turn connect into other regional or national ISPs. FIG. 1B illustrates an example of connections to an ISP. In the example, home user end stations may connect via modems over dial-up lines to an ISP's router or remote access server (RAS). This data link often runs the PPP (Point-to-Point Protocol) which encapsulates and delivers packets to the ISP's site. Business user end systems may connect to the ISP through leased lines such as T1 lines or T3 lines depending on bandwidth requirements for example. Other examples of typical connection options between home or business users and an ISP include ISDN, T1, fractional T1, various optical media, and xDSL. ISPs may also offer tunnel mode or transport mode services that help businesses set up virtual private networks (VPNs) between remote end stations and virtual dial-up services for remote and mobile end stations.

The ISP serves as a conduit for information transmitted between the end stations in the home and other end stations connected to the Internet.

A virtual circuit is a dedicated communication channel between two end stations on a packet-switched or cell-relay network. ATM, Frame Relay, and X.25 are all different types of virtual circuit based networking technologies. A virtual circuit follows a path that is programmed through the intermediate stations in the network.

There are permanent and switched virtual circuits. A permanent virtual circuit (PVC) is permanent in the sense that it survives computer reboots and power cycles. A PVC is established in advance, often with a predefined and guaranteed bandwidth. A switched virtual circuit (SVC) is "switched" in the sense that it can be created on demand analogous to a telephone call. Both PVCs and SVCs are "virtual" circuits in that they typically are not allocated their own physical links (e.g. wires), but share them with other virtual circuits running across the same physical links.

"Tunneling" is one mechanism for building higher layer networks on top of an underlying virtual network. Tunneling has already gained acceptance in the industry and several technologies are either in operation or under development. Some of the tunneling protocols used in IP networks for example include L2TP, GRE, PPTP, and L2F. There are many other Tunneling technologies used in IP and other protocols.

Referring to FIGS. 2A-2B, there are shown network graphs representing two illustrative networks. Network A is represented by three nodes (NA1, NA2, and NA3), and three links (LA1, LA2, and LA3). Network B is represented by four nodes (NB1, NB2, NB3, and NB4) and four links (LB1, LB2, LB3, and LB4). As used herein, the term node may represent any end station or intermediate station, and the term link means any connection between nodes. If these are physical nodes and links, Networks A and B are physically isolated from each other. If these are virtual (circuit) links which actually depend on a shared physical medium, then

the two (virtual) networks are said to be virtually isolated from each other.

Illustrative Networks A and B each may be part of different network domains. Independent administrative control may be exercised over each of the Network A and B domains, for example, through the configuration and management of intermediate stations such as bridges and routers.

Referring to FIGS. 2A and 2B, it will be appreciated that the independent administration of the Network A and Network B domains may result in incompatible policies as between the two domains. This is not a problem provided that the domains remain isolated from each other. Referring to FIG. 3, however, there is shown a network graph of Network C which comprises Networks A and B joined by link LJ. The isolation between Networks A and B, whether physical or virtual, is lost when they are joined in Network C. This joining of the two Networks A and B may create challenges to the administration of combined Network C. For example, despite the joining of the two networks, there still may be a need to apply different or even conflicting policies to each of Networks A and B. In essence, the administrative challenge is to maintain the administrative integrity of the Network A domain and the administrative integrity of the Network B domain despite the fact that both of these networks are part of Network C and are no longer physically isolated from each other.

FIG. 4 is an illustrative drawing of a segment of a single physical medium capable of carrying multiple information flows, each in its own virtual circuit (or channel). The physical medium may for instance be a cable or a wire or an optical fiber. The segment shown is carrying four independent information flows on four different virtual circuits; VC1, VC2, VC3, and VC4. These virtual circuits, for example, may be implemented using X.25, ATM, Frame Relay, or some other virtual circuit (or channelized) service.

FIG. 5 is an illustrative drawing representing an example of two virtual networks (VN1, and VN2) each made up of two independent network segments (VN1.1 and VN1.2 for VN1, and VN2.1 and VN2.2 for VN2). All segments connect to shared physical network resources. In this example, the shared network resources of FIG. 5 provide a virtual circuit service. A virtual circuit connection to an end station or intermediate station connection to a virtual circuit is called a virtual channel connection (VCC). VN1 connects at VCC1 and VCC4; and VN2 connects at VCC2 and VCC3. The shared network resources also provide virtual circuit service that connect VCC1 and VCC4 so as to join VN1.1 and VN1.2 into VN1 and so as to join VN2.1 and VN2.2 into VN2.

FIG. 6 is an illustrative drawing that provides additional details of some of the physical constituents of the virtual networks of FIG. 5. An intermediate station labeled VN1.1.VCC1 in VN1 connects segment VN1.1 to the VC service at VCC1. An intermediate station labeled VN1.2.VCC4 in VN1 connects segment VN2 to the VC service at VCC4. The VC service connects VCC1 to VCC4, linking VN1.1 to VN1.2 at the virtual circuit level. More specifically, physical media segments PM2, PM1 and PM5 and intermediate stations IS-A and IS-B provide the requisite physical infrastructure upon which the virtual circuit connection linking VN1.1 and VN1.2 is carried. This first virtual circuit connection serves as a network link between the VN1.1.VCC1 and VN1.2.VCC4 intermediate stations, to create one virtual network from the two segments VN1.1 and VN1.2.

Similarly, VCC2 and VCC3 are connected by the virtual circuit service, which connects intermediate stations

VN2.1.VCC2 and VN2.2.VCC3, joining the VN2.1 and VN2.2 segments to form the virtual network labeled VN2. More particularly, physical media segments PM4, PM1 and PM3 and intermediate stations IS-A and IS-B provide the virtual connection linking VN2.1 and VN2.2. The second virtual circuit connection serves as a network link between the VN2.1.VCC2 and VN2.2.VCC3 intermediate stations, to create one virtual network from the two segments VN2.1 and VN2.2.

FIG. 7 is an illustrative drawing shows the logical or higher level view of the two virtual networks VN1 and VN2 of FIGS. 5 and 6. It will be appreciated from the view of FIG. 6 that they share physical resources, and it will be appreciated from the view of FIG. 7 that they are logically or virtually separate.

In the illustrative example of FIG. 8, two virtual networks are layered on top of a third virtual network. The sharing of a common set of physical or virtual network resources by several virtual networks increases the challenges of maintaining isolation and security of the individual virtual networks. Nevertheless, end user requirements for information resources, technology advances, economics, politics, and regulations surrounding the networking industry are driving commercial, private and government entities to share common physical and virtual network infrastructure. Therefore, there are ever increasing demands imposed upon network administrators, and vendors of networking equipment.

In the illustrative drawing of FIG. 8, three separate network domains intersect at node IN1: i) that of the Internet itself (including or subsuming that of the underlying VC service supporting the Internet); ii) that of private virtual network VN1; and iii) that of private virtual network VN2. This intersection of three network domains creates the potential for the kinds of administration and policy challenges discussed above. It will be noted that these networks are represented by different network "clouds" that symbolize the multifarious nodes and links in each of the networks.

The illustrative drawing of FIG. 8 illustrates an example of building two virtual networks on top of another virtual network similar to the previous example in FIGS. 5, 6 and 7. As before, the virtual networks being overlaid are each composed of two segments. Using a tunneling protocol or some other higher layer (layer 3 or above) mechanism, connections are made between nodes IN1.1 and IN1.2 to form a link to tie the two segments of VN1 together. This link is shown as T1 in FIG. 9 and 10. Link T2 is similar, formed between nodes IN2.1 and IN2.2, to tie the two segments of VN2 together. The logical view of the two virtual networks in FIG. 9 is shown in FIG. 10, which bears a very strong resemblance to FIG. 7. The important difference to note between the examples is that in FIG. 7 a layer 2 VC network was used as the underlying network shared resources, and in FIG. 10 another virtual network was used as the underlying network shared resources; specifically, a tunneled service across the Internet. Thus, it will be appreciated that different virtual networks can be formed in different layers using the same underlying physical (or virtual) network resources.

Connections are established between nodes at the edge of the segments where they interface or connect to the shared (Internet) resources which are analogous to the virtual circuits in FIGS. 5, 6, and 7. These may be tunneled connections, or connections built using some other (connection-less) technology.

If we assume T1 and T2 are tunnels, the network databases of IN1.1, IN1.2, IN2.1, and IN2.2 would be aug-

mented with data structures to manage the tunneling protocol at those endpoints, and the links made up from the tunnels. The network database of IN1.1 of FIG. 8 is depicted in FIG. 11 which highlights the "Tunneling Database" and the "IP Database".

Network Database Organization

If we examine the information in the network database for IN1, we will see that it should include configuration and policy information for three separate domains. Furthermore, since the information from the three domains must all coexist in the same physical device, there should be some way to structure the information and control its usage, such that the IN1 device operates correctly in all three domains.

If all information for the device IN1 were stored in one monolithic form as is done conventionally, in addition to all the policies for each domain, inter-domain policies would also be required to ensure that information should be kept private to its own domain.

The illustrative drawing of FIG. 12 is a generalized drawing of a conventional monolithic structure for a database that can be used to implement node IN1 of FIG. 7. The drawing depicts, in a conceptual fashion, an example of the typical organization of information within such a device.

The illustrative device includes a first interface attached to VN1.1, a second interface attached to VN2.2 and a third interface attached to the Internet as the shared network resources. To illustrate the complexities in the database design, assume that both the virtual networks being overlaid on the Internet are also (private) IP networks (internets).

Therefore all three networks/domains operate using the IP protocol, each having its own independent IP information to be stored in IN1's network database.

The database includes information such as rules used to articulate and implement administrative policies. The policies as articulated in the information and rules, for example, may include security rules, restrictions on access and dynamic routing protocols. In this illustrative router, the policy information and policy rules used to control the layer 3 IP protocol routing for all three networks are included in a single monolithic database.

However, as explained above, different network domains may have different or perhaps even conflicting policies. In order to provide at least some degree of isolation, additional and complicated "inter-domain" policy mechanisms must be added to manage the conflicts between policies on similar data from different domains. These mechanisms are configured and managed by an administrative authority. The dotted lines in FIG. 12 represent the points at which these inter-domain policy mechanisms would be introduced. The policies would attempt to divide the monolithic network database of node IN1 into three separate domain-specific sections. These dotted lines indicate that separation policy mechanisms are implemented, to provide at least some isolation of the information pertaining to VN1 from the information pertaining to VN2, and also from the information pertaining to the Internet (i.e. shared network resources).

It will be appreciated that the complexity and difficulty in defining and administering the policy mechanisms used to achieve isolation can be great. There is potential for a wide range of policies to be defined between domains. Everything in the spectrum from almost complete openness and sharing of all information between domains, to the other extreme of not sharing anything at all are possible. Certain pieces of a domain's database may want to be kept private (e.g. access

control policy configuration), while other parts are shared to some extent (e.g. summarized routing and addressing information). The types of data, and the extent to which they can all be shared, are all subject to restriction through definition of inter-domain policies.

If we consider each boundary between a pair of domains (i.e. each dotted line through the network database of IN1 in FIG. 12) as a separate policy object, it will also be appreciated that the number of policy objects increases much faster than the number of domains. If D is the number of domains, then P , the number of policy objects can be calculated approximately as:

$$P = (D(D-1))/2$$

Thus, the number of policy objects increases approximately as (a proportion of) the square of the number of domains. In other words, the number of policy objects ordinarily increases much faster than the number of domains, especially as the number of domains gets large.

Another challenge in the administration of virtual networks arises because home or business end station users may wish to change the nature of their connections to the network from time to time. For instance, an end user may wish to utilize a more expensive higher bandwidth connection for business use and a less expensive lower bandwidth connection for home or personal use. Alternatively, for instance, an end user may wish opt to receive a video transmission on a higher bandwidth connection while still receiving other transmissions on lower bandwidth connections. An end user may even wish to change the ISP that he or she uses. Unfortunately, these changes often require intervention by a network administration authority to change the higher level binding between the end user station and the network. More specifically, the binding (or association) between the layer 2/1 virtual circuit service and a layer 3 intermediate device is 'hard', not dynamic, and the higher layer interface generally must be reconfigured by a network administrator to change the binding.

Thus, there has been a need for improved organization of network domain databases and improvements in the ability of a network user to change network domain. The present invention meets these needs.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1A is a generalized diagram of a multi-protocol bridge/router.

FIG. 1B is an illustrative example of the topology of and connections.

FIGS. 2A and 2B are network graphs of two illustrative example networks.

FIG. 3 is a network graph of an illustrative network in which the networks of FIGS. 2A and 2B are joined.

FIG. 4 is an illustrative drawings of a segment of a single physical medium capable of carrying multiple information flows which in its own virtual circuit (or channel).

FIG. 5 is an illustrative drawings of two virtual network each made up of two independent segments.

FIG. 6 is an illustrative drawings that provides additional details of some of the physical constituents of the virtual networks of FIG. 5.

FIG. 7 is an illustrative drawings which shows the logical or higher level view of the two virtual network VN1 and VN2 of FIGS. 5 and 6.

FIG. 8 is an illustrative drawings that shows that the Internet can provide the shared network resources of FIGS. 5 and 6.

FIG. 9 is an illustrative drawings that shows tunneling through the Internet to provide the shared resources of FIGS. 5 and 6.

FIG. 10 is a logical or high level view of the two virtual networks of FIG. 9.

FIG. 11 is a generalized illustrative drawing of the organization of node IN1 to achieve tunneling.

FIG. 12 is a conceptual drawing of one possible router configuration that can be used to implement intermediate node IN1 of FIG. 7.

FIG. 13 is a generalized block diagram of a network device that instantiates multiple virtual network machine routers in electronic in accordance with one embodiment of the invention.

FIG. 14 is a generalized block diagram of a network device that instantiates a virtual network machine with multiple layer 2 sub-interface data structures and multiple layer 3 interfaces and binding data structures that associate layer 2 sub-interface data structures and layer 3 interfaces.

FIG. 15 is a generalized block diagram of the network device of FIG. 14, except that one binding data structure has been removed and another binding data structure has been created.

FIG. 16 is a generalized block diagram of a network device that implements a virtual network machine router and a virtual network machine bridge.

FIG. 17 is a generalized block diagram of the network device as in FIG. 16, except that one binding data structure has been removed and another binding data structure has been created.

FIG. 18 is a generalized block diagram of the network device of FIG. 14, except that one binding data structure has been eliminated and another binding data structure has been created.

FIG. 19 is a generalized block diagram of a network device which comprises a computer which instantiates multiple virtual machines in accordance with an embodiment of the invention.

FIG. 20 is generalized block diagram of the network device of FIG. 19 except that one binding data structure has been removed and another binding data structure has been created.

FIG. 21 is a generalized block diagram of a subscriber management system in accordance with a presently preferred embodiment of the invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

The present invention comprises a novel apparatus and method for managing operation of network devices that can operate in multiple virtual network domains. The following description is presented to enable any person skilled in the art to make and use the invention, and is provided in the context of particular applications and their requirements.

Various modifications to the preferred embodiments will be readily apparent to those skilled in the art, and the generic principles defined herein may be applied to other embodiments and applications without departing from the spirit and scope of the invention. Thus, the present invention is not intended to be limited to the embodiments shown, but is to be accorded the widest scope consistent with the principles and features disclosed herein.

Virtual Network Machines

A Virtual Network Machine (VNM) as the term is used herein to describe the collection of processes and mecha-

nisms that operate on a network device to implement the functions of a node in a virtual network. The preferred embodiment for the VNM is as a set of computer programs and related data structures encoded in electronic memory of a network device and used to operate on information, consuming some portion of a network device's computer and memory storage capacity. The functionality of a virtual network machine can be that of a router, bridge or switch, depending on what is configured in its network database. The native resources of a network device include its processor(s), memory, I/O, communication hardware and system software. The native resources of a network device, for example, may include peripheral devices or even a server computer which may, for instance, provide information about end user privileges or virtual network configurations.

Referring to the illustrative drawing of FIG. 13, there is shown a generalized block diagram of a new structure for the network database of node IN1 from FIGS. 8 and 12 in accordance with one-embodiment of the invention that supports creation of multiple virtual network machines. In this case, the networks device IN1 supports three virtual network machines VNM0, VNM1 and VNM2. In the embodiment of FIG. 13, assuming again that all three virtual networks operate using the IP protocol, each virtual machine implements the functionality of an IP router, each operating in its own network domain. Each virtual network machine is allocated a portion of the device's native resources. Each virtual network machine runs the IP protocol stack. Each virtual network machine stores its address, policy and control information separately from the others. Thus, each virtual network machine can operate independently of the other virtual network machines, even though it shares native computer resources with the other virtual network machines. This virtual network machine based organization of information therefore provides greater isolation between network domains.

Each virtual machine has its own network database that contains its control information. VNM0 has a network database that causes it to operate as a router that routes information within the Internet network domain. VNM1 has a network database that causes it to operate as a router that routes resource information within network domain VN1. VNM2 has a network database that causes it to operate as a router that routes resource information within network domain VN2.1. *High Speed Networks, TCP/IP and ATM Design Principles*, by William Stallings, Prentice Hall, 1998 provides detailed discussion of router functions and the functions of other network devices.

The VNMs of FIG. 13 may employ multiple different kinds of layer 1 (physical) media to attach to one or more networks. In a presently preferred embodiment, these physical connections include ATM OC-3c/STM1, ATM DS-3/E3, DS-3 Clear Channel, HSSI and 10/100 Base-2 T TX. Resource information is transmitted across these physical connections such as phone lines, DSL or ADSL for example to and from VNM0, VNM1 and VNM2 using layer 2 (data link) protocols. There are layer 2 LAN (local area network) technology and layer 2 WAN (wide area network) technology protocols. Examples of LAN technologies include Ethernet and IEEE 802.3, Fast Ethernet, Token Ring and Fiber Distributed Data Interface. Examples of WAN technologies include Asynchronous Transfer Mode (ATM), Frame Relay, X.25, Point-to-Point (PPP), Integrated Services Digital Network (ISDN) and High-Level Data Link Control (HDLC). Intermediate stations communicate with each other using layer 3 protocols. Layer 3 protocols include Internet Protocol (IP), AppleTalk and Inter Packet Exchange (IPX). Thus,

for example, VNM0, VNM1 and VNM2 each employ one or more layer 3 protocols to communicate with other stations of the network(s) to which they are attached.

Thus, the three virtual machines and the different network domains associated with them are isolated from each other in the network device intermediate station of FIG. 13, and the task of exercising administrative control can be simplified significantly. Since there is no monolithic database that must be maintained to control information transfers across all of the networks to which the three VNMs are attached, the task of administering each database is simplified.

The virtual network machine based organization also simplifies the administration, lowering the cost of operating all three networks. The organization of information along network domain boundaries eliminates the notion of information from two domains residing under a single monolithic structure, and thereby eliminates the need to define inter-domain policies to manage the separation of information within a monolithic database structure. The separation policy mechanisms represented by the dotted lines cutting through the database of FIG. 12 are gone, and a whole set of administrative chores disappears with them. There will be no need to define the complicated inter-domain policies, and no cost associated with administering them. The amount of information that needs to be configured by the administrators is greatly reduced in size and complexity using this method of database organization.

Other benefits can be realized through greater efficiencies in the implementation of such network devices that are possible with this method of network database organization. Further efficiencies are realized through the elimination of the complicated inter-domain policies in virtually all functions of the device. Essentially, each of the virtual machines VNM0, VNM1 and VNM2 operates a separate/independent network device, performing networking functions its own domain.

Dynamic Binding

The drawing of FIG. 14 shows another illustrative embodiment of the invention. The IP network device of FIG. 14 implements a router that includes three network interfaces NIF3-0, NIF3-1 and NIF3-2. The network device also has a layer 1/2 connection to an Ethernet service. The network device also has a layer 1/2 connection to a virtual circuit service. An Ethernet service sub-interface data structure Eth1 provides the layer 2 Ethernet connection such as sub-interface data structure provides the layer 2 VCC1 connection. For example, the VCC1 sub-interface data structure of FIG. 14 may be kept in a table that identifies all virtual circuit connections, each defining the encapsulation protocol, the packet or cell, data compression technique and the particular layer 2 protocol used on that circuit. The Ethernet sub-interface data structure may include the Ethernet address of the local connection and other parameters to control transmit and receipt of information on the Ethernet segment. A binding data structure B3-0 binds the Ethernet sub-interface data structure to NIF3-0. A binding data structure B3-2 binds the VCC1 sub-interface data structure to NIF3-2. The Ethernet and VCC1 sub-interface data structures are labeled with the prefix "sub" because they are layer 2 constructs which are below the layer 3 interface constructs in the ISO scheme.

Referring to FIG. 14, binding data structure B3-0 establishes a layer 2/3 connection between the Ethernet sub-interface data structure and NIF3-0, and binding data structure B3-2 establishes a layer 2/3 connection between VCC1

sub-interface data structure and IF3-2. Binding data structure B3-0 causes information transferred across the Ethernet connection to be processed through to NIF3-0. An IP Forwarding/Routing database controls routing of the information out the correct interface. Binding data structure B3-2 causes the information transferred across the VCC1 connection to be processed through NIF3-2.

The VCC1 sub-interface data structure instantiates a virtual circuit connection to the network device of FIG. 14. A virtual circuit connection such as that in FIG. 14 can be created in accord with any of several technologies. A sub-interface data structure like that in FIG. 14 stores the network device's identity of the virtual circuit attached to it. Many virtual circuits can be established across a single physical connection, and many virtual circuits can be connected to a single network device.

FIG. 15 depicts the same intermediate station as in FIG. 14, except the binding B3-0 has been eliminated, and binding B3-1 has been created. Binding B3-1 associates the Ethernet sub-interface data structure Eth-1 with interface NIF3-1. Interface NIF3-2 remains bound to the sub-interface data structure VCC1. The interface NIF3-0 is not bound to any layer 2 construct. It should be noted that an unbound interface construct generally would represent a mis-configuration in a typical earlier intermediate station.

FIG. 16 depicts yet another illustrative embodiment of the invention. The network device of FIG. 16 implements an IP router function and a bridging function. The router includes two interfaces NIF4-1 and NIF4-2. The bridge includes a bridge interface BR4-0. A network database that implements the bridge function includes a list of network stations reachable through each of the bridge's interfaces. The network device also has a layer 1/2 connection to an Ethernet service. The network device also has a layer 1/2 connection to a virtual circuit service VCC1. An Ethernet service sub-interface data structure Eth1 provides information concerning the Ethernet connection such as a VCC1 sub-interface data structure provides information concerning the VCC1 connection. A binding data structure B4-0 binds the Ethernet sub-interface data structure to NIF4-0. A binding data structure B4-2 binds the VCC1 sub-interface data structure to NIF4-2. NIF4-1 is unbound.

FIG. 17 depicts the same network device as in FIG. 16, except the binding B4-0 has been eliminated, and binding B4-1 has been created. Binding B4-1 associates the Ethernet sub-interface data structure with interface NIF4-1 of virtual router VM4. Interface NIF4-2 remains bound to the sub-interface data structure VCC1. The interface BR4-0 is not bound to any layer 2 construct. These changes in binding effectively redefines the service available on the Ethernet segment from a bridged or layer 2 service, to a routed or layer 3 service. In a presently preferred embodiment of the invention, these bindings can be changed without reconfiguration of any other interface construct or circuit construct. In a typical earlier intermediate station, the bindings between the higher and lower layers are implicit, and a change in the implicit bindings applied to the bridge and router interface constructs typically would have required a modification of these interface constructs. A present embodiment of the invention does not require such modification.

FIG. 18 depicts the same network device as in FIG. 14, except the binding B3-0 has been eliminated and binding B3-2A has been created. Binding B3-2A associates the Ethernet sub-interface data structure with the NIF3-2 interface. Binding B4-2 associates the VCC1 sub-interface data structure with NIF3-2. Interfaces NIF3-0 and NIF3-1 are

unbound. This change in bindings causes both the Ethernet and the virtual circuit lower layer services to be associated with a single higher layer 1P construct, NIF3-2.

FIG. 19 shows a network device which comprises a computer which instantiates multiple virtual network machines VNM5 and VNM6. VNM5 implements IP router functionality. It includes network interfaces NIF5-0 and NIF5-1. VNM6 also implements IP router functionality. It includes two interfaces NIF6-0 and NIF6-1. The network device of FIG. 19 has two layer 1/2 connections to a virtual circuit service. Sub-interface data structure VCC1 instantiates one of the connections to the device. Sub-interface VCC2 instantiates the other connection to the device. A binding data structure B5-0 binds the VCC1 sub-interface data structure to NIF5-0 of VNM5. A binding data structure B6-2 binds the VCC2 sub-interface data structure to interface NIF6-1 of VNM6. VNM5 and VNM6 each use the IP protocol suite to communicate with other stations of the network(s) to which they are attached.

FIG. 20 depicts the same network device as in FIG. 19, except the binding B5-0 has been eliminated and binding B6-0 has been created. The binding B6-0 data structure associates VCC1 sub-interface data structure with NIF6-0 of VNM6. Binding data structure B6-1 binds sub-interface data structure VCC2 to NIF6-1. Neither of the VNM5 interfaces NIF5-0 and NIF5-1 are bound.

In FIGS. 14 to 20, bindings are shown as data structures connected to other data structures by line segments. In one preferred embodiment, the line segments each represent a pair of bi-directional pointers; the first pointer points from the binding to the higher or lower layer data structures and the second is opposite the first, pointing from the higher or lower layer data structure to the binding data structure. Alternatively, the binding could be implemented as indices or identifiers in a table, for example. Dynamic binding is accomplished by creating and or deleting binding data structures and or changing the values of the pointers or indices so they operate on different data structures. It will be appreciated that actual changing of the bindings can be accomplished through entries in a command line interface to the network device or automatically by snooping the information flow through the device, for example.

The illustrative drawing of FIG. 21 is a generalized block diagram of a subscriber management system in accordance with a presently preferred embodiment of the invention. A subscriber is a user of network services. The system includes a computer with layer 1/2 connections to subscriber end stations and with layer 1/2 connections to network devices that provide access to other networks.

The system can form a multiplicity of layer 1/2 subscriber end station connections. In a present embodiment, the layer 1/2 connections to subscriber end stations include virtual circuit connections. The system memory stores a multiplicity of sub-interface data structures that instantiate the multiplicity of virtual circuit connections through which subscriber end stations communicate with the subscriber management system.

The system instantiates in memory a plurality of virtual network machines. Each VNM of the embodiment of FIG. 21 implements the functionality of a router. There are nine illustrative VNM routers shown in FIG. 21 labeled VNM₁-VNM₉. Each VNM router includes interfaces in its database. Each VNM router runs at least one layer 3 protocol suite. Each VNM router may run one or more adaptive routing algorithms. The interfaces of each VNM router provide access to a network that is isolated from the

networks accessed through the interfaces of the other VNM routers. For example, the interface to VNM-4 provides layer 3 access to the network that includes ISP#2. The interface to VNM-5 provides layer 3 access to the network that includes Corporate-Private-Network#A. The interface to VNM-6 provides layer 3 access to the network that includes ISP#4. The networks with ISP#2, Corporate-Private-Network#A and ISP#4 are isolated from each other. The databases associated with VNM-4, VNM-5 and VNM-6 to control access to networks across these respective interfaces. Each of these three VNM databases can be administered separately.

In operation a subscriber might establish a point-to-point connection with the subscriber management system. A server that runs software that runs authentication, authorization and accounting protocols (AAA) searches for a record that identifies the user. Authentication is the process of identifying and verifying a user. For instance, a user might be identified by a combination of a username and a password or through a unique key. Authorization determines what a user can do after being authenticated, such as gaining access to certain end stations information resources. Accounting is recording user activity. In the present embodiment, AAA involves client software that runs on the subscriber management system and related access control software that runs either locally or on a remote server station attached to the network. The present embodiment employs Remote Authentication Dial-In User Service (RADIUS) to communicate with a remote server. An example of an alternative AAA protocol is Terminal Access Controller Access Control System (TACACS+). RADIUS and TACACS+ are protocols that provide communication between the AAA client on a router and access control server software.

The subscriber record includes information concerning the network to which the subscriber's virtual circuit connection should be bound. Typically, the subscriber will employ a PVC. Based upon the information in the subscriber record, a binding data structure, like that described in reference to FIGS. 14 to 20, will be created to associate the sub-interface data structure that instantiates the PVC in the subscriber management system memory with the interface to the VNM router that accesses the network identified for the subscriber in the subscriber record.

Moreover, the subscriber record may provide multiple possible binding options for the subscriber. For instance, the subscriber may specify the creation of a binding that is which is to be employed during business hours and which binds the subscriber to VNM-5 which provides layer 3 network access to the Corporation-Private-Network#. The same record may specify another binding which is to be employed only during non-business hours and which binds to VNM#4 which provides layer 3 network access to ISP#2. Thus, the bindings can be changed. They are dynamic.

Various modifications to the preferred embodiments can be made without departing from the spirit and scope of the invention. Thus, the foregoing description is not intended to limit the invention which is described in the appended claims in which:

What is claimed is:

1. A computer implemented method comprising:

routing Internet Protocol (IP) packets within a first Internet Service Provider's (ISP's) domain from a single network device with a first database, the first database including addresses of the first ISP's domain; and
routing IP packets within a second ISP's domain from the single network device with a second database, the

second database being separate from the first database and including addresses of the second ISP's domain.

2. The computer implemented method of claim 1, wherein the first database also includes control and policy information for the first ISP's domain and the second database includes control and policy information for the second ISP's domain.

3. The computer implemented method of claim 1 further comprising connecting a subscriber to the first ISP's domain with an authentication, authorization and accounting protocol.

4. The computer implemented method of claim 1 further comprising:

routing IP packets within a corporation's domain from the single network device with a third database, the third database being separate from the first and second databases, wherein said third database includes addresses of the corporation's domain.

5. The computer implemented method of claim 1 further comprising:

providing the corporation administrative control of the third database, but not the first and second databases; providing the first ISP administrative control of the first database, but not the second and third databases; and providing the second ISP administrative control of the second database, but not the first and third databases.

6. The method of claim 1 further comprising routing the packets within the first ISP's domain with a global database that includes globally known addresses if the packets cannot be routed within the first ISP's domain with the first database.

7. A memory having a set of one or more programs stored thereon to cause a single network device to perform operations comprising:

maintaining a first database separately from a second database in the single network device, the first database having addresses for a first Internet Service Provider's (ISP's) domain and the second database having addresses for a second ISP's domain;

routing Internet Protocol (IP) packets within the first ISP's domain from the single network device with the first database; and

routing IP packets within the second ISP's domain from the single network device with the second database.

8. The memory of claim 7 further comprising providing access to a subscriber to the first ISP's domain with an authentication, authorization, and accounting protocol.

9. The computer implemented method of claim 7 further comprising:

maintaining a third database separately from the first and second databases, wherein the third database has addresses for a corporation's domain; and

routing IP packets within the corporation's domain with the third database from the single network device.

10. The computer implemented method of claim 9 further comprising:

providing the first ISP administrative control of the first database, but not the second or third databases;

providing the second ISP administrative control of the second database, but not the first or third databases; and

providing the corporation administrative control of the third database, but not the first or second databases.

11. The memory of claim 7 wherein the set of one or more programs cause the single network device to perform operations further comprising:

17

maintaining a third database separately from the first and second databases, wherein the third database has addresses of a backbone; and routing IP packets within the first ISP's domain with the third database if they cannot be routed with the first database. 5

12. A single network device comprising:

an electronic memory having

a first database of network addresses of a first network domain that is administered by a first Internet Service Provider (ISP); 10

a second database of network addresses of a second network domain that is administered by a second ISP, the second database being isolated from the first database; and

a set of one or more processors to execute a set of instructions that cause the single network device to route a first set of packets of the first network domain with the first database and to route a second set of packets of the second network domain with the second database. 20

13. The single network device of claim 12 wherein the packets are IP packets.

14. The single network device of claim 12 wherein the packets are layer 2 packets. 25

15. The single network device of claim 12 wherein the first set of packets are transmitted from a subscriber of the first ISP and the second set of packets are transmitted from a subscriber of the second ISP. 25

16. The single network device of claim 12 further comprising:

the electronic memory further having

a third database to store network addresses of a third network domain that is administered by a corporation, the third database being isolated from the first and second databases; and

the set of processors to execute the set of installations to further cause the single network device to route a third set of packets of the third network domain with the third database. 30

17. The single network device of claim 12 further comprising:

the electronic memory having a third database of addresses of a network provider that is administered by the network provider, the third database being isolated from the first and second databases; and

the set of processors to execute the set of instructions that further cause the single network device to route the first set of packets with the third database if they cannot be routed with the first database. 35

18. A method comprising:

routing packets for a first set of subscribers with a first virtual router, and routing packets for a second set of subscribers with a second virtual router, the first and second virtual routers being isolated from each other within a single network device, the first set of subscribers subscribing to a first Internet Service Provider (ISP) and the second set of subscribers subscribing to a second ISP; 40

providing administrative control of the first virtual router, which includes a first network database, used by the first virtual router to route packets, of network device addresses within the first ISP's domain and control and policy information for the first ISP's domain, to the first ISP; and

providing administrative control of the second virtual router, which includes a second network database, used

18

by the second virtual router to route packets, of network device addresses within the second ISP's domain and control and policy information for the second ISP's domain, to the second ISP, wherein the first ISP does not have administrative control of the second network database and the second ISP does not have administrative control of the first network database.

19. The method of claim 18 wherein the packets are layer 2 packets.

20. The method of claim 18 wherein providing administrative control of the first virtual router comprises allowing the first ISP to modify the first network address database and the control and policy information governing the first ISP's domain. 15

21. The method of claim 18 further comprising: providing a network provider administrative control of a global virtual router including a global network database in the single network device.

22. The method of claim 18 further comprising: routing packets for a third set of subscribers with a third virtual router; and providing administrative control of the third virtual router, which includes a third network database of network device addresses within a corporation's domain and control and policy information for the corporation's domain, wherein the corporation has administrative control of the third virtual router but not the first and second virtual routers. 20

23. The method of claim 22 further comprising providing a network provider access to the first, second and third network databases and a global network database, said global network database being in said single network device. 25

24. The method of claim 18 wherein the packets are layer 3 packet. 30

25. The method of claim 18 further comprising: connecting the first and second set of subscribers to the single network device in accordance with a authorization, authentication and accounting protocol. 35

26. An electronic memory encoded with a set of instructions, which when executed on a single network device, cause said single network device to perform operations comprising:

creating a plurality of collections of processes and mechanisms for implementing router functionality, each of the plurality of collections of processes and mechanisms operating on a different network database including addresses and control and policy information; 40

separately storing the network database of each of the plurality of collections of processes and mechanisms; and

each of the plurality of collections of processes and mechanisms routing packets within a different administrative domain with its network database and in accordance with its control and policy information. 45

27. The electronic memory of claim 26 wherein the packets are layer 2 packets.

28. The electronic memory of claim 26 wherein each of the plurality of collections of processes and mechanisms runs its own IP stack. 50

29. The electronic memory of claim 26 wherein at least one of the different administrative domains is administered by an Internet Service Provider.

30. The electronic memory of claim 26 wherein at least one of the different administrative domains is administered by a corporation. 55

31. A single network device comprising:
 plurality of virtual network machines that are individually isolated, each of the plurality of virtual network machines to route packets within a different administrative domain with
 a network address database for the different administrative domain, and
 control and policy information for the different administrative domain;
 a first port to transmit and receive said packets to and from subscribers; and
 a second port to transmit and receive said packets to and from the Internet.

32. The single network device of claim 31 wherein the plurality of virtual network machines are virtual IP routers.

33. The single network device of claim 31 wherein the different administrative domains include an Internet Service Provider domain and a corporate domain.

34. The single network device of claim 31 further comprising a third port to transmit and receive a second set of packets to and from a corporate network domain.

35. A single network device comprising:
 communication hardware;
 a set of one or more processors coupled with the communication hardware; and
 an electronic memory coupled with the communication hardware and the set of processors, the electronic memory encoded with a set of instructions to cause the set of processors to,
 host a first virtual router that includes a first network database of network device addresses within a first Internet Service Provider's (ISP's) domain and control and policy information for the first ISP's domain, and
 host a second virtual router, isolated from the first virtual router, that includes a second network database of network device addresses within a second ISP's domain and control and policy information for the second ISP's domain,
 route packets for a first set of subscribers with the communication hardware and the first virtual router, wherein the first set of subscribers subscribe to the first ISP,
 route packets for a second set of subscribers with the communication hardware and the second virtual router, wherein the second set of subscribers subscribe to the second ISP,
 provide the first ISP administrative control of the first virtual router but not the second virtual router, and
 provide the second ISP administrative control of the second virtual router, but not the first virtual router.

36. The single network device of claim 35 further comprising the electronic memory to host a global network database.

37. The single network device of claim 35 further comprising:
 the electronic memory to host a third virtual router that includes a third network database of network device addresses within a corporation's domain and control and policy information for the corporation's domain, and the set of instructions to further cause the set of processors to,
 route packets for a third set of subscribers with the communication hardware and the third virtual router, and
 provide the corporation administrative control of the third virtual router, but not the first and second virtual routers.

38. The single network device of claim 35, wherein the set of instructions further cause the set of processors to switch packets for a third set of subscribers with the communication hardware and the first virtual router.

39. The single network device of claim 35 wherein the packets are layer 3 packets.

40. A network comprising:
 a set of one or more networks;
 a set of one or more end stations communicating, for a first set of one or more subscribers of a first Internet Service Provider (ISP) and for a second set of one or more subscribers of a second ISP, packets;
 a single network access device coupled between the set of networks and the set of end stations, the single network access device having, communication hardware;
 an electronic memory coupled with the communication hardware, the electronic memory having stored therein,
 a first network database, controllable for administration by the first ISP but not the second ISP, including network device addresses and control and policy information for the first ISP,
 a second network database, controllable for administration by the second ISP but not the first ISP, including network device addresses and control and policy information for the second ISP, wherein the fast network database and the second network database are isolated from each other; and
 a set of one or more processors, coupled with the communication hardware and the electronic memory, routing said packets being communicated for the first set of subscribers with the communication hardware and a first virtual router that includes the first network database, and routing said packets being communicated for the second set of subscribers with the communication hardware and a second virtual router that includes the second network database.

41. The network of claim 40 further comprising the electronic memory having stored therein a global network database.

42. The network of claim 40 fiber comprising:
 the set of one or more networks including a virtual network of a corporation;
 the set of end stations communicating, for a third set of one or more subscribers of the corporation, packets;
 the electronic memory having stored therein a third network database, controllable for administration by the corporation but not the first ISP nor the second ISP, including network device addresses and control and policy information for the corporation; and
 the set of processors routing said packets being communicated for the third set of subscribers with the communication hardware and a third virtual router that includes the third network database.

43. The network of claim 40 further comprising the set of processors switching packets for a third set of one or more subscribers with the communication hardware and the first virtual router.

44. The network of claim 40 wherein the packets are layer 3 packets.

45. An electronic memory encoded with a set of instructions, which when executed on a single network device, cause said single network device to perform operations comprising:

routing packets for a first set of subscribers with a first virtual router, and routing packets for a second set of subscribers with a second virtual router, the first and second virtual routers being isolated from each other within the single network device, the first set of subscribers subscribing to a first Internet Service Provider (ISP) and the second set of subscribers subscribing to a second ISP;

providing administrative control of the first virtual router, which includes a first network database of network device addresses within the first ISP's domain and control and policy information for the first ISP's domain, to the first ISP; and

providing administrative control of the second virtual router, which includes a second network database of network device addresses within the second ISP's domain and control and policy information for the second ISP's domain, to the second ISP, wherein the first ISP does not have administrative control of the second network database and the second ISP does not have administrative control of the first network database.

46. The electronic memory of claim 45, wherein the operations further comprise:

routing packets for the first and second set of subscribers with a third virtual router, the third virtual router including a global network database.

47. The electronic memory of claim 45 wherein the operations further comprise:

routing packets for a third set of subscribers with a third virtual router, the third virtual router being isolated from the first and second virtual router within the single network device; and

providing a corporation administrative control of the third virtual router, which includes a third network database of network device addresses of the corporation and control and policy information for the corporation, wherein the corporation has administrative control of the third virtual router but not the first and second virtual routers.

48. The electronic memory of claim 45, wherein the operations further comprise:

switching packets for a third set of one or more subscribers with the first virtual router.

49. A method in a single network device comprising: creating a plurality of collections of processes and mechanisms for implementing router functionality, each of the plurality of collections of processes and mechanisms operating on a different network database including addresses and control and policy information; separately storing the network database of each of the plurality of collections of processes and mechanisms; and

each of the plurality of collections of processes and mechanisms routing packets within a different administrative domain with its network database and in accordance with its control and policy information.

50. The computer implemented method of claim 49 wherein at least one of the plurality of collections of processes and mechanisms switching packets within its administrative domain with its network database.

51. The computer implemented method of claim 49 wherein each of the plurality of collections of processes and mechanisms runs its own IP stack.

52. The computer implemented method of claim 49 wherein at least one of the different administrative domains is administered by an Internet Service Provider.

53. The computer implemented method of claim 49 wherein at least one of the different administrative domains is administered by a corporation.

54. The computer implemented method of claim 49 wherein the different network database of each of the plurality of collections of processes and mechanisms includes addressing information, control information, and policy information.

55. A single network device comprising:
a set of one or more processors; and
an electronic memory coupled with the set of processors, the electronic memory having a set of instructions to cause the set of processors to,
create a plurality of collections of processes and mechanisms, each of the plurality of collections of processes and mechanisms to operate on a different network database including control and policy information, and to route packets within a different administrative domain with its network database and in accordance with its control and policy information, and
separately store the different network database of each of the plurality of collections of processes and mechanisms.

56. The single network device of claim 55 further comprising at least one of the collections of processes and mechanisms to switch packets with its network database and in accordance with its control and policy information.

57. The single network device of claim 55 wherein the set of instructions further cause the set of processors to independently run an Internet Protocol stack for each of the plurality of collections of processes and mechanisms.

58. The single network device of claim 55 wherein at least one of the different administrative domains is administered by an Internet Service Provider.

59. The single network device of claim 55 wherein at least one of the different administrative domains is administered by a corporation.

60. The single network device of claim 55 wherein the different network database of each of the plurality of collections of processes and mechanisms is to include addressing information, control information, and policy information.

61. A network comprising:
a set of one or more networks;
a set of one or more end stations communicating packets with the set of networks; and
a single network device coupled between the set of networks and the set of end stations, the single network device having a plurality of collections of processes and mechanisms, each of the plurality of collections of processes and mechanisms, operating on a different network database including addresses and control and policy information, wherein the network database operated on by each of the collection of processes and mechanisms is stored separately, and

routing packets within a different administrative domain with its network database and in accordance with its control and policy information.

62. The network of claim 61 further comprising at least one of the plurality of collections of processes and mechanisms switching packets with its network database and in accordance with its control and policy information.

63. The network of claim 61 each of the plurality of collections of processes and mechanisms independently running an IP stack.

64. The network of claim 61 wherein at least one of the different administrative domains is administered by an Internet Service Provider.

65. The network of claim 61 wherein at least one of the different administrative domains is administered by a corporation.

66. A network comprising:

a set or one or more networks;
a set of one or more end stations communicating packets with the set of networks, and

a single network device coupled between the set of networks and the set of end stations, the single network device having,

a first virtual network machine transmitting certain of said packets for a first subscriber in accordance with a first network database of a first administrative domain, the first database having addressing and policy information of the first administrative domain, and

a second virtual network machine, which is isolated from the first virtual network machine, transmitting certain packets for a second subscriber in accordance with a second network database, the second network database having addressing and policy information for a second administrative domain.

67. The network of claim 66 wherein the first administrative domain is administered by a corporation.

68. The network of claim 66 wherein the first administrative domain is administered by an Internet Service Provider.

69. The network of claim 66 wherein the first administrative domain provides a first service and the second administrative domain provides it second service.

70. The network of claim 66 wherein the addressing information in the first database includes layer 2 addressing information.

71. The network of claim 66 wherein the addressing information in the first database include layer 3 addressing information.

72. A single network device comprising:

a first set of one or more ports to receive IP packets from a first and second set of one or more subscribers;

a second set of one or more ports to transmit IP packets over a first network domain;

a machine-readable medium having stored therein a set of instructions to cause the single network device to, instantiate a first and second virtual router, which are virtually-independent but share a set of physical resources within the single network device,

the first virtual router to route within a second network domain, which is layered upon the first network domain, WP packets from the first set of subscribers using a first network database that includes IP addresses, control and policy information defined for the second network domain, and the second virtual router to route within a third network domain, which is layered upon the first network domain and shares the first network domain's physical resources with the second network domain, IP packets from the second set of subscribers using a second network database that includes IP addresses and control and policy information defined for the third network domain,

maintain separation between the first and second network databases so as to avoid management of inter-domain policies, wherein avoidance of inter-domain policies eases administrative tasks,

provide for independent administration of the first and second network databases, wherein independent administration maintains administrative integrity of the first and second network databases.

73. The single network device of claim 72 wherein the first set of subscribers are subscribers of a corporate network and the second set of subscribers are subscribers of an Internet service provider.

74. The single network device of claim 72 wherein the first network domain is a layer 3 network domain and the second and third network domains are layer 3 network domains.

75. The single network device of claim 72 further comprising the set of instructions to cause the single network device to tunnel IP packets of the first set of subscribers between separate physical locations of a virtual private

network with the first virtual router.

76. The single network device of claim 72 further comprising the set of instructions to cause the single network device to process the first and second set of subscribers in accordance with an authorization, authentication and accounting protocol.

77. A single network device comprising:

a first set of one or more ports to receive IP packets from subscribers;

a second set of one or more ports to transmit IP packets over a first network domain;

a machine-readable medium having stored therein a set of instructions to cause the single network device to, instantiate different virtual routers for different network domains, which are layered upon the first network domain, using separate unshared inter-domain policy free, independently administrable network databases, wherein each of the separate unshared inter-domain policy free, independently administrable network databases includes IP addresses, control and policy information defined for its one of the different network domains, and

route IP packets of different ones of the subscribers using those of the virtual routers for the different ones of the network domains to which those subscribers currently belong.

78. The single network device of claim 77 wherein the subscribers includes subscribers of a corporate network and subscribers of different Internet service providers.

79. The single network device of claim 77 wherein the first network domain is a layer 3 network domain and the different network domains layered upon the first network domain are layer 3 network domains.

80. The single network device of claim 77 further comprising the set of instructions to cause the single network device to tunnel IP packets of certain of the subscribers between separate physical locations of a virtual private network with the one of the different virtual routers to which the certain of the subscribers belong.

81. The single network device of claim 77 further comprising the set of instructions to cause the single network device to process the subscribers in accordance with an authorization, authentication and accounting protocol.

* * * * *



US006594704B1

(12) **United States Patent**
Birenback et al.

(10) **Patent No.:** US 6,594,704 B1
(45) **Date of Patent:** Jul. 15, 2003

(54) **METHOD OF MANAGING AND USING
MULTIPLE VIRTUAL PRIVATE NETWORKS
IN A ROUTER WITH A SINGLE ROUTING
TABLE**

(75) Inventors: **Michael Birenback, Acton, MA (US);
Mark Duffy, Lexington, MA (US);
Sanjay Krishnan, Arlington, MA (US)**

(73) Assignee: **Quarry Technologies, Burlington, MA
(US)**

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: 09/461,957

(22) Filed: **Dec. 15, 1999**

(51) Int. Cl.⁷ **G06F 15/173**

(52) U.S. Cl. **709/238; 709/242; 709/245;
709/249; 711/206; 711/207**

(58) Field of Search **709/238, 245,
709/249, 242; 711/206, 207**

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,386,413 A * 1/1995 McAuley et al. 370/392
5,606,650 A * 2/1997 Kelley et al. 345/428

5,809,563 A *	9/1998	Yamada et al.	711/207
6,160,811 A *	12/2000	Partridge et al.	370/359
6,205,488 B1 *	3/2001	Casey et al.	709/238
6,226,751 B1 *	5/2001	Arrow et al.	370/351
6,339,595 B1 *	1/2002	Rekhter et al.	370/392
6,426,955 B1 *	7/2002	Gossett Dalton, Jr. et al.	370/401
6,438,612 B1 *	8/2002	Ylonen et al.	709/249
6,526,056 B1 *	2/2003	Rekhter et al.	370/392
2003/0018876 A1 *	1/2003	Zahir et al.	711/206

* cited by examiner

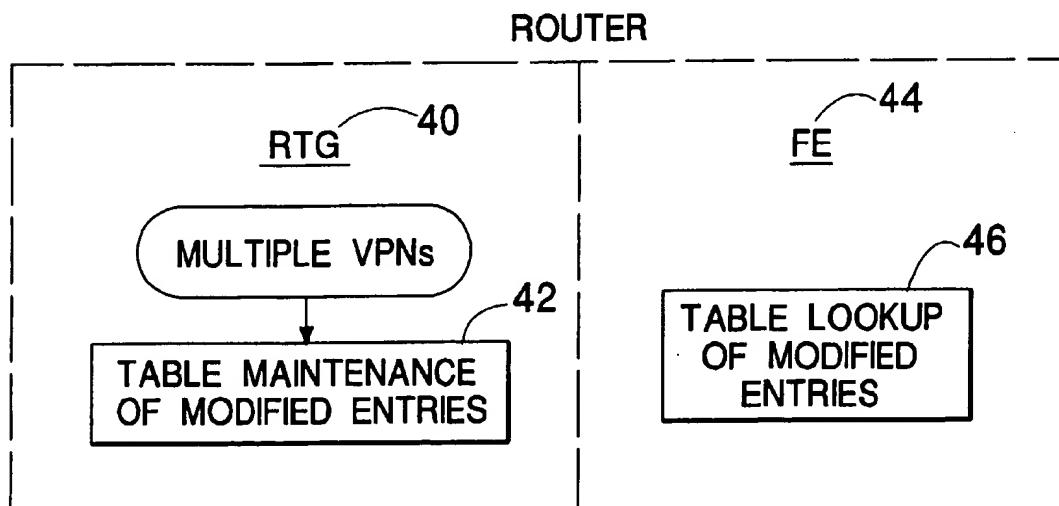
Primary Examiner—Nabil El-Hady

(74) *Attorney, Agent, or Firm*—Fish & Richardson P.C.

(57) **ABSTRACT**

A method of maintaining multiple routing tables within a global table of a network router including the steps of providing the router with a route table generator to maintain the global table, and a forwarding engine to perform table lookups, the performing of table lookups in one of the multiple routing tables of the global table using a key, the key representing an index into the one of said multiple routing tables within said global table, the key created by combining a virtual private network identification (VPN ID) and an address prefix, and maintaining a plurality of table entries in the global table by using the key as a pointer to one of said table entries.

13 Claims, 1 Drawing Sheet



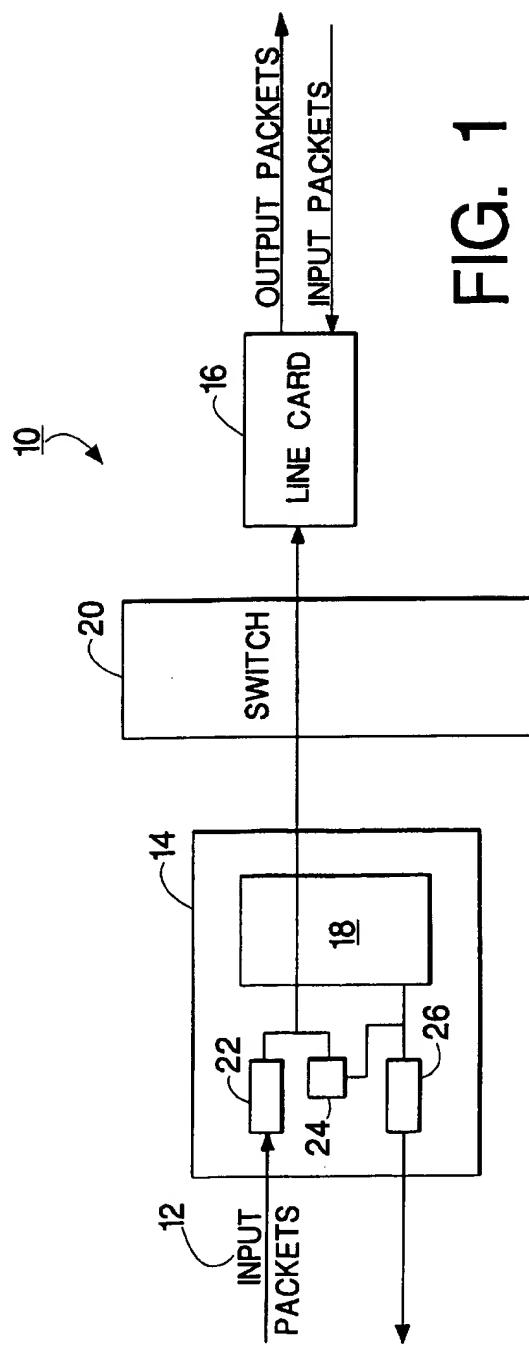


FIG. 1

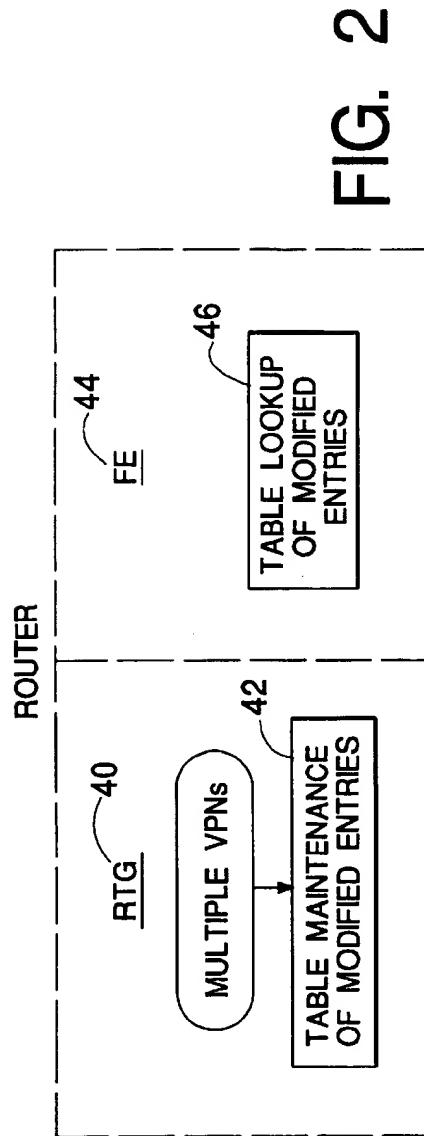


FIG. 2

**METHOD OF MANAGING AND USING
MULTIPLE VIRTUAL PRIVATE NETWORKS
IN A ROUTER WITH A SINGLE ROUTING
TABLE**

FIELD OF THE INVENTION

The present invention relates to the field of telecommunications, and more particularly to an improved system and method of searching.

BACKGROUND OF THE INVENTION

As is known, the Internet is a packet network, where each data packet is passed from a network device referred to as a router to the next in the chain, until it reaches its destination. Typically, as a packet moves through the network, each router performs packet forwarding decisions independent of other routers and other packets. Internet Protocol (IP) is the networking language of the Internet, and routers are the principal engines of IP. IP routers have to perform a per-packet lookup in a routing table to determine where IP packets are to be forwarded.

Within a router, a route entry essentially consists of two parts—an address prefix to next hop mapping (at the IP layer) and the knowledge of what link is available to that next hop. These two pieces of information are sufficient to route packets. A packet enters the router with a destination address in the IP header. The router performs a lookup of its routing table with this destination address as the key. The routing table itself is essentially an address prefix to next hop mapping. Hence, given a destination address, the router selects an address prefix entry in the routing table to determine the next hop router that this packet must be handed to. The routing table entry also indicates what link to send the packet out on (to this next hop).

Typically the router receives millions of packets per second and for efficient and fast forwarding, the need for a high speed destination prefix lookup is essential. The destination address in a given packet may match multiple prefixes in the routing table and so the router is required to find the most specific match from a large number of patterns.

Several schemes for better router data throughput and faster packet forwarding have been proposed in recent years in order to perform Internet Protocol (IP) lookups in an optimized manner. Most applications require the retrieved information be associated with the best matching prefix (BMP), if any, of the input string. For example, if the string "MART" is the input string to a database, and the database holds information associated with the strings "M", "MA", and "MAL", the best matching prefix to "MART" is the string "MA", and the information associated with "MA" should be returned. Note that "M" is also a prefix of "MART", but "MA" is a better (i.e. longer) prefix than "M". BMP searching is typically performed by a database having a hierarchical, tree-like structure. This type of database is often called a trie. A trie database allows both exact matching (i.e. searching for a string that is exactly equal to the input string) as well as best prefix.

IP lookups that use BMP can be performed in either software or hardware. An example of a software approach, as mentioned above, is to involve a tree-based method using Patricia, i.e., radix, trees, or binary trees. An example of a hardware approach is to involve the use of Content Addressable Memories (CAMs).

A typical router has two main forwarding functions, i.e., address lookup (to lookup a destination address) and mes-

sage switching (sending the packet to the correct output link). The problem of accomplishing address lookups at gigabit speeds is ever increasing. With the emergence of high-speed links, like OC-48 and higher (An OC-48 link can 5 deliver packets at up to 2.5 million packets per second assuming an average of 1,000 bit per packet) to the router, it is essential that packet forwarding keep up with this rate. Otherwise packet forwarding (route lookup, etc.) could act 10 as a bottleneck essentially degrading the performance of the router.

Complicating matters is greater usage of virtual private networks. A virtual private network (VPN) is a private data network that makes use of the public telecommunication infrastructure, maintaining privacy through the use of a 15 tunneling protocol and security procedures. A virtual private network can be contrasted with a system of owned or leased lines that can only be used by one company. The idea of the VPN is to give the company the same capabilities at much lower cost by using the shared public infrastructure rather 20 than a private one. Phone companies have provided secure shared resources for voice messages. A virtual private network makes it possible to have the same secure sharing of public resources for data. Companies today are looking at 25 using private virtual networks for both extranets and wide-area intranets.

An extranet is a private network that uses the Internet protocols and the public telecommunication system to 30 securely share part of a business's information or operations with suppliers, vendors, partners, customers, or other businesses. An extranet can be viewed as part of a company's intranet that is extended to users outside the company.

An intranet is a private network that is contained within 35 an enterprise. It may consist of many interlinked local area networks and also use leased lines in the wide area network. Typically, an intranet includes connections through one or 40 more gateway computers to the outside Internet. The main purpose of an intranet is to share company information and computing resources among employees. An intranet can also be used to facilitate working in groups and for teleconferences. An intranet may use TCP/IP, HTTP, and other Internet 45 protocols and in general look like a private version of the Internet.

As is well known, implementation of multiple VPNs in a 45 single router has typically complicated address lookup since each VPN must have its own routing table, necessitating a large amount of table space within a single router to implement these individual routing tables. Also, dividing the available table space into dedicated areas for each VPN creates fragmentation and therefore inefficient use of the 50 available space.

What is needed is a new and improved system and method 55 of managing multiple VPN routing tables within a single shared space to maintain efficiency and speed of lookups, and correspondingly, efficiency and speed of packet forwarding.

SUMMARY OF THE INVENTION

In accordance with the principles of the present invention 60 a method of maintaining multiple routing tables within a global table of a network router is provided including the steps of providing the router with a route table generator to maintain the global table, and a forwarding engine to perform table lookups, the performing of table lookups in one 65 of the multiple routing tables of the global table using a key, the key representing an index into the one of said multiple routing tables within said global table, the key created by

combining a virtual private network identification (VPN ID) and a packet destination address, and maintaining a plurality of table entries in the global table by using the key as a pointer to one of said table entries.

BRIEF DESCRIPTION OF THE DRAWINGS

The novel features believed characteristic of the invention are set forth in the appended claims. The invention itself, however, as well as features and advantages thereof, will be best understood by reference to the detailed description of specific embodiments which follows, when read in conjunction with the accompanying drawings, wherein:

FIG. 1 is a block diagram of an exemplary network router; and

FIG. 2 is a block diagram of route table generator and forwarding engine of the router of FIG. 1 in accordance with the principles of the present invention.

DESCRIPTION OF THE PREFERRED EMBODIMENT(S)

Referring now to FIG. 1, a simplified block diagram of an exemplary router 10 is shown to illustrate a data processing path for a stream of input packets 12 entering from a line card 14 and exiting from a line card 16. The exemplary router 10 consists of multiple line cards, 14 and 16, each supporting one or more network interfaces (not shown). Line card 14 is shown to include a Forwarding Engine (FE) 18. Line card 14 and line card 16 are plugged into a high-speed switch 20. By way of example, when a packet 22 arrives at the line card 14, its header 24 is removed and passed to the forwarding engine 18. The Forwarding Engine 18 reads the header 24 to determine how to forward the packet 22 and then updates the header 24 and sends the updated header and its forwarding instructions back. The line card 14 integrates the new header with the rest of the packet 22 and sends the entire packet 22 through the switch 20 to the line card 16 for transmission. A Network Processor (NP) 26 provides basic management functions such as link up/down management and generation of Forwarding Engine routing tables for the router 10.

The router 10 contains a routing table on the Network Processor (NP) 26; the routes are contained in an IP Forwarding Database. The IP Forwarding Database contains next hop IP address, interface to get to the next hop, etc. The majority of Forwarding decision is done in the FE 18. The Forwarding Engine 18 is a hardware module that performs lookups (on the incoming packet) of the routing table. The routing table is maintained by software running on the Network Processor 26 called the Route Table Generator (RTG). Routers typically save table size by summarizing several address entries by using a single prefix entry. With such an arrangement, best matching prefix (BMP) is used to solve the problem of possibly multiple prefixes matching a given address.

As mentioned previously, the term virtual private network (VPN) is used in many different ways and can mean different things to different people. To best understand what a VPN is, it helps to also have a clear definition of both a private network and a public network.

A private network dedicates all network components (leased or owned) to a single customer. A public network shares one or more network components among multiple customers. A virtual private network (VPN) uses a public network, along with encryption, tunneling (encapsulation), and authentication, to achieve the same level of security and

privacy as a private network. Virtual private networks are used in two primary ways. First, VPNs can connect two networks. This is typically referred to as a LAN-to-LAN VPN or a site-to-site VPN. Second, a remote access VPN can connect a remote user to a network. A remote access VPN replaces the need for a private network. In a VPN, one organization's data travels over the same Internet as everyone else's; no additional physical network is created. But, because the data is encrypted and only authenticated and authorized users can access the data, a virtual network that is private and secure exists. In fact, VPNs on the Internet are thought by some to be more secure than WANs that do not employ data encryption.

When implementing a single VPN within the router 10, for example, a single routing table is maintained, allowing use of a single address space to create and maintain the table. When implementing more than one VPN within the router 10, a virtual router, which is one of the elements of VPN, is created for each supported VPN. A virtual router is one of (possibly) multiple logical routers manifested within one physical router unit. Within the virtual router, prior to the present invention, each of the VPNs would necessitate the creation and maintenance of its own separate routing table for destination address lookups. The implementation of multiple routing tables results in fragmentation of memory space, such that memory space in the machine must be devoted to the routing table of one virtual router or another. Since memory space is finite, flexible use of memory space is one motivation behind the present invention.

Hence, if a router is supporting five VPNs, it needs six routing tables, i.e., one routing table for each VPN and one routing table for a global prefix table. The RTG is capable of maintaining multiple routing tables, restricted only by memory limitations. Configuration of the IP protocol stack assigns virtual private network identifications (VPN IDs) to each VPN. Each route entry in a router is part of a VPN ID, where VPN ID=0 represents the global prefix space in the internet.

The present invention allows the creation and management of a single, integrated routing table for support of multiple VPNs. Referring to FIG. 2, a high level block diagram of the operation and implementation is shown. As mentioned above, an Route Table Generator (RTG) 40 creates and maintains a single routing table 42, i.e., the RTG 40 adds, deletes, and modifies entries within the table 42. Entries within the table 42 are typically prefixes, and standard methods of creating prefix entries are employed. However, each of the table entries utilized in the present invention is placed in the table 42 by an index resulting from a hashed combination of the prefix with a VPN ID, more fully described below.

A forwarding engine (FE) 44 (sometimes referred to as a forwarding process) is a hardware/software module that performs one or more hash lookups 46 (on behalf of the incoming packet) of the routing table 42 in accordance with any one of a number of table lookup algorithms. When a packet is received, a VPN ID is associated with the packet based on the context in which the packet was received (using standard techniques well known in this art). Next, a portion 60 of a destination address is combined with the VPN ID, as is done by the RTG 40 in maintaining the table 42, described fully below; this combination of VPN ID and a portion of the destination address is then hashed to obtain an index into the table whereupon best matching prefix (BMP) lookups 46 can occur utilizing standard BMP algorithms. Thus, the present invention manages one table which can effectively service multiple VPNs by defining keys into a table, through the use

of VPN ID and prefix, thus resulting in managing a number of sub-tables, each supported within the single table.

The RTG 40 adds entries to routing tables corresponding to particular virtual routers, each of which is hashed. The present invention combines the address prefix with VPN ID, which yields the hashed location within the combined table, for the specific table entry.

When a packet is received, the FE performs a number of hashed lookups in accordance with the route lookup algorithm utilized. At each hash lookup, the combination of VPN ID and destination address are used as search keys.

As can be seen above, the RTG uses a specific combination of VPN ID and prefix as a key to entries within the table, allowing the table to be maintained. This same specific combination of VPN ID and a portion of the destination address of a received packet is used by the FE to find appropriate entries in the table so that a BMP algorithm can find the appropriate match and result in forwarding the packet to the correct destination within the correct VPN.

The present invention utilizes the following method in the RTG to maintain the forwarding prefix table and by the FE to find the appropriate table entries associated with a specific VPN. The method involves performing an arithmetic function of the VPN ID and the destination address. Although a specific example will be described here which involves manipulating the VPN ID and destination address with a bitwise OR, one skilled in this art will see that any suitable arithmetic operation may be performed which will result in the same intended result. Thus, by way of one illustrative example, in the RTG, a key into the table is created as follows: Each route entry to be added has an associated VPN ID which identifies the virtual router to which the route entry belongs. The VPN ID is an 8-bit value, which is left shifted 24 bits. This shifted VPN ID is then combined with a prefix by using a bitwise OR. The resulting value is used as a key into the table of prefixes for purposes of adding, changing, or deleting a route entry.

In the FE, a packet is received. The VPN ID, which is based on the context of where the packet has arrived from, is determined. The VPN ID is an 8-bit value, which is left shifted 24 bits. Using a bitwise OR of the shifted VPN ID and a portion of the destination address of the received packets results in a key into the forwarding prefix table.

Having described a preferred embodiment of the invention, it will now become apparent to those skilled in the art that other embodiments incorporating its concepts may be provided. It is felt therefore, that this invention should not be limited to the disclosed invention, but should be limited only by the spirit and scope of the appended claims.

What is claimed is:

1. A method comprising:

providing a router with a route table for routing incoming packets to multiple virtual private networks (VPNs); generator to maintain a global table, and a forwarding engine to perform table lookups, the performing of the table lookups in the global table using a key, the key representing an index pointing into the global table, the key generated by combining a virtual private network identification (VPN ID) of a packet and a portion of the packet; and

maintaining a plurality of table entries in the global table by using the key as a pointer to one of the table entries.

2. The method according to claim 1 wherein identifying the VPN is determined from a context of where the packet arrived from.

3. The method according to claim 1 wherein generating the key pointing into the global table of entries comprises

performing a mathematical operation on the VPN ID and the portion of the packet, the portion of the packet being a segment of a destination address of the packet.

4. A method for routing a packet in a virtual router servicing multiple private networks comprising:

receiving a packet on an input port of a virtual router; identifying a virtual private network identification (VPN ID) of the packet;

identifying a portion of the packet;

generating a key that points into a global table of entries, the key combining the portion of the packet and the VPN ID;

performing a search within the global table of entries by using the key as an offset into a specific location of the global table to determine a best matching prefix (BMP); and

routing the packet to an output port of the virtual router in accordance with the results of performing the search.

5. A method comprising:

maintaining a plurality of entries in a global routing table supporting multiple virtual private networks (VPNs), each of the entries using an access key for access, the access key being a combination of a VPN identification (ID) and a prefix;

receiving a plurality of packets into a router;

identifying a VPN ID of each of the packets;

identifying a portion of a destination address of each of the packets;

combining the VPN ID and the portion of the destination address to produce a key;

performing a search using the key as an index into the global routing table; and

routing each of the packets in accordance with the search results.

6. The method according to claim 5 wherein identifying a VPN ID of each of the packets comprises identifying a context of each of the packet.

7. The method according to claim 5 wherein combining the VPN ID and the portion of the destination address to produce a key comprises performing an arithmetic operation on the VPN ID and the portion of the destination address.

8. The method according to claim 7 wherein performing an arithmetic operation comprises:

left shifting the VPN ID; and

performing a mathematical operation on the shifted VPN ID and the portion of the destination address.

9. A method comprising:

in a network, receiving a packet in a routing system servicing a plurality of virtual private networks (VPNs);

generating a key from a VPN identification (VPN ID) of the received packet; and a portion of the received packet; and

determining an entry in a global routing table having port addresses of next hops for said plurality of VPNs using the key.

10. The method according to claim 9, wherein the portion of the received packet is a segment of a destination address of the packet.

11. The method according to claim 9, wherein the entry is a port address of a next hop.

12. The method according to claim 9, further comprising sending the packet out to a port corresponding to the entry in the table.

13. A method comprising

receiving a packet on an input port of a router; supporting 5 a plurality of virtual private networks (VPNs);

generating a key that points into a global table of entries representing addresses for said plurality of VPNs, the

key combining a portion of the packet and a virtual private network identification (VPNs) VPN ID; determining a match of the key against a location in the global table; and sending the packet to an output port of the router corresponding to the match.

* * * * *

UNITED STATES PATENT AND TRADEMARK OFFICE
CERTIFICATE OF CORRECTION

PATENT NO. : 6,594,704 B1

Page 1 of 1

DATED : July 15, 2003

INVENTOR(S) : Michael Birenbach, Mark Duffy and Sanjay Krishnan

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

Title page.

Item [12], "Birenback" should be spelled -- Birenbach --.

Item [75], Inventors:, "Birenback" should be spelled -- Birenbach --.

Signed and Sealed this

Twenty-fifth Day of November, 2003



JAMES E. ROGAN

Director of the United States Patent and Trademark Office